

Offenes Parlament Documentation

version 1.1

Ben Freundorfer, Florian Cech, Markus Hametner

December 13, 2017

Contents

Welcome to the documentation for Offenes Parlament	1
Installation/Rollout Instructions	1
Installation Notes	1
Installation instructions with Vagrant	1
Prerequisites	1
Setup	1
Resetting the database	1
Creating a Model-Diagram	2
Initial scraping	2
ElasticSearch and Re-Indexing	2
Technical Documentation	2
General Information	2
Structure - What is Where?	2
Frontend Code	3
Scraping: Scrapy Spiders and Extractors	3
Structure	3
Spiders	3
Saving/Updating the models	5
Keyword parameters	5
Extractors	5
Search Provider: Haystack, Elasticsearch	6
Basics	6
Re-Indexing	6
SearchViews	6
Facet Only Search	9
Paging	9
Fieldsets	9
Single Result Search	10
Indices	21
Django Admin Interface	21
Manually trigger scraping	21
Import-Export	21
Frontend	21
Tech	21
Building	22
Grunt too slow?	22
Subscriptions via Email	22
Basics	22
Views and URL-Schemes	22
Email Templates	23

API	23
General & Structure	23
Implementation	23
Caveats	25
Existing Scrapers	25
Inquiries	25
General notes about the structure of the inquiries	25
Scraper Structure	26
Model structure	27
Example Inquiry	27
Design	27
Design	28
Fonts	28
Logo	28
Colors	28
Grid	29
Text	29
Tables	29
Links	29
Icons	29
Indices and tables	30
Community	30

Welcome to the documentation for Offenes Parlament

An open-data framework for the public data of the Austrian Parliament

Most of this documentation is technical in nature and sketches out implementation details, design patterns and installation/running instructions.

This version of the document is licensed as Creative Commons CC-BY (Attribution).

Installation/Rollout Instructions

Installation Notes

This section aims to briefly sketch out the process of installing the app for development.

Installation instructions with Vagrant

The suggested method includes Vagrant, a virtualization tool aimed at developers.

Prerequisites

- [Vagrant](#)
- [Vagrant Hostmanager Plugin](#)
- [Virtualbox](#)

Setup

1. Clone the github repository (duh)
2. Navigate into the project dir `cd OffenesParlament`
3. Setup and run the vagrant VM `vagrant up`. All requirements will be installed automatically inside the VM which may take a few minutes the first time.
4. The script might ask you for your password as it will add `offenesparlament.vm` pointing to this VM to your hosts-file. It also automatically creates a django superuser `admin` with password `admin`.
5. Log in to the running VM with `vagrant ssh`
6. For the initial scraping instructions see below
7. Run the server inside the VM (0.0.0.0 lets the server respond to requests from outside the VM - ie your physical machine where you probably run your browser)

```
cd offenesparlament
python manage.py runserver 0.0.0.0:8000
```

8. If you work on client files that have to be compiled (CSS, JS) you have to run grunt as well. ATM we have the tasks `dev` and `reloading`. `dev` watches and regenerates files when their sources change. (Remember that sources also change when you do a git pull and generated client files aren't committed to git) And `reloading` does that and uses [Browsersync](#) to reload your browser when files change

```
cd /vagrant
grunt dev
```

9. To exit and shutdown the VM run

```
exit
vagrant halt
```

Resetting the database

In case you need to reset the database (delete all migrations, flush the db content, recreate all objects etc.), run these commands in the django project folder 'offenesparlament':

```
bin/clear_db.sh
```

Creating a Model-Diagram

It's possible to view the current database model residing in the op_scraper app by calling:

```
bin/graph_models.sh
```

A png-image will be generated as `ignore/models.png`.

Initial scraping

There are currently four available scrapers, which should initially run in this order:

1. llp (legislative periods)
2. persons (for instance *Rudolf Anschöber* <http://www.parlament.gv.at/WWER/PAD_00024/index.shtml>)
3. administration (for instance, 'Faymann II' and all the Persons having a mandate for that administration)
4. pre_laws (for instance *Buchhaltungsagenturgesetz, Änderung (513/ME)*)
5. laws_initiatives (for instance *ÖBIB-Gesetz 2015 (458 d.B.)*)

To run a scraper, use the following command:

```
python manage.py scrape crawl <scraper_name>
```

for instance:

```
python manage.py scrape crawl persons
```

The law_initiatives scraper also has an additional parameter to define, which legislative period to scan; per default, it scrapes the periods from XX to XV. This can be overridden this way:

```
python manage.py scrape crawl -a llp=21 laws_initiatives
```

to only scrape that period. Careful though: scraping of periods before the 20th legislative period is not possible as of yet (since there are no machine-readable documents available).

ElasticSearch and Re-Indexing

For now, reindexing (or updating the index, for that matter), is only done manually. To have all data indexed, just run:

```
python manage.py rebuild_index
```

for a full rebuild (wipes the indices first), or:

```
python manage.py update_index
```

to perform a simple update. For this to succeed, make sure ElasticSearch is up and running.

Technical Documentation

General Information

This section outlines the general setup of OffenesParlament.at.

Structure - What is Where?

OffenesParlament.at is a (set of) Django applications. It's roughly divided into two parts: the *Scraping* part, which takes care of data aggregation and parsing of the website of the [Austrian Parliament](#), and the *Presentation* part, which presents the collected data in the form of a searchable web application.

The *Scraping* part can be found in the subfolder `offenesparlament/op_scraper`, whereas the *Presentation* part is largely gathered in the folder `offenesparlament/offenesparlament`.

The search engine parts transcend each of those projects, with the search views being located in `/offenesparlament/offenesparlament/search_views.py`, but the `search_indexes` being located in `/offenesparlament/op_scraper/search_indexes.py`.

In future, the Email-Subscription service will also be located in both parts, given that we will have to offer a subscription logic in the webapp itself (with a set of views to facilitate that), but also trigger the sending of those emails via the scraper upon changing the database.

Frontend Code

All sources for the frontend code (JS and CSS) are located in `client/` and split between `scripts` and `styles`. We use CoffeeScript and the React framework for client code. To generate JS and CSS from the sources we use grunt (`Gruntfile.coffee` is in the root dir).

All generated files are put in `offenesparlament/offenesparlament/static/` by grunt.

Scraping: Scrapy Spiders and Extractors

This section describes the scraping setup and processes.

Structure

The scraper is located in the subfolder `/offenesparlament/op_scraper`. It contains the Django models (cf. `/offenesparlament/op_scraper/models.py`), some *admin views* and *admin dashboard* adaptations, as well as the [Austrian Parliament](#) scraper itself, situated at `/offenesparlament/op_scraper/scraper/parlament`.

A scrapy scraper consists of a set of *spiders* - a single process capable of scanning, parsing and injecting data from a website into a database. Currently, the following spiders exist:

- **laws_initiatives**: Scrapes the Laws and Government Initiatives as found on [this page](#)
- **pre_laws**: Scrapes laws that are still in the pre-parliamentary process, as shown on [this list](#)
- **llp**: Scrapes the list of available legislative periods
- **persons**: Scans 'Parlamentarier' as found [here](#)
- **administrations**: A secondary spider that also scans Persons, this time focused on their mandates as part of specific administrations, as shown [in here](#)
- **statements**: Scrapes Debates and DebateStatements. Requires *llp* and *persons* for lookup.
- **Inquiries**: Scrapes Anfragen (Inquiries) and Beantwortungen. Requires *Persons*.

Each of those spiders inherits from `BaseSpider` (cf. `/offenesparlament/op_scraper/scraper/parlament/spiders/__init__.py`), which offers some generic methods to be used by different spiders.

Besides the spiders themselves, which handle getting the response from the subsite of `parlament.gv.at` and creating the django objects based on the scraped data, the *Extractors* (to be found at `/offenesparlament/op_scraper/scraper/parlament/resources/extractors`) do the actual heavy lifting of translating the raw html data into meaningful, structured data (mostly in the form of dictionaries and lists) by using XPATH expressions.

Spiders

Spiders are the managing part of the scraping process. At the bare minimum, a spider consists of an Constructor (the `__init__` method), which is responsible for populating the `self.start_urls` list with all the web-adresses to be scanned, as well as a parse method, which gets to be called with the response from each of the entries in the `self.start_urls` list. Furthermore, each spider **must** have a member variable `name` set, which will identify it for the command line calls.

The following is a simple example or code skeleton of a spider:

```

# -*- coding: utf-8 -*-
from parliament.settings import BASE_HOST
from parliament.spiders import BaseSpider
from parliament.resources.extractors.example import EXAMPLE_EXTRACTOR
from ansicolor import green

from op_scraper.models import ExampleObject

class ExampleSpider(BaseSpider):
    BASE_URL = "{}/{}/{}".format(BASE_HOST, "/WWER/PARL/")

    name = "example"

    def __init__(self, **kw):
        super(ExampleSpider, self).__init__(**kw)

        self.start_urls = [self.BASE_URL]

    def parse(self, response):

        data_sets = EXAMPLE_EXTRACTOR.xt(response)

        for data_set in data_sets:
            item, created = ExampleObject.objects.update_or_create(
                name=data_set['name'],
                defaults=data_set
            )
            item.save()

            if created:
                self.logger.info(u"Created ExampleObject {}".format(
                    green(u'{}'.format(data_set['name']))))
            else:
                self.logger.info(u"Updated Legislative Period {}".format(
                    green(u'{}'.format(data_set['name']))))
        ))

```

Not all database/django objects can be fully extracted through a single page. For instance, the *Person* objects need to be discovered through one of the abovementioned lists, but their details can only be extracted from a secondary person detail page. To accomodate this, scrapy's callback functions can be used like this person spider skeleton:

```

def parse(self, response):

    # Parse person list
    # [...]

    callback_requests = []
    for p in person_list:
        # Create Detail Page request
        req = scrapy.Request(person_detail_page_url,
                             callback=self.parse_person_detail)
        req.meta['person'] = {
            'reversed_name': p['reversed_name'],
            'source_link': p['source_link'],
            'parl_id': parl_id
        }
        callback_requests.append(req)

    return callback_requests

```



```
def parse_person_detail(self, response):

    person = response.meta['person']

    # Parse Person detail page
    # [...]
```

In the above example, the spider will start making secondary requests to retrieve the detail pages, and call the `parse_person_detail` with the responses. As shown above, the request for the secondary page contains a member variable `meta` that can be used to transfer already created data to the secondary response to continue working with the same person and provide some continuity.

Saving/Updating the models

Currently, the spiders do not need to take care of versioning the changes they scrape; since the page needs to be requested and scraped already to be able to determine if there were any changes, the spiders should simply update existing objects or create new ones where necessary. Since the `OffenesParlament.at` app also employs `django.reversion` to version the changes to the database, it can be possible to trace changes to the objects via versions rather than during the scraping process itself, although this is not yet implemented due to the fact that the email-subscription service hasn't been realized yet.

Keyword parameters

To specify additional (optional) keyword parameters for the spiders, the `__init__` method accepts a `kw` parameter, which contains a dictionary of keys and values supplied from the commandline. For instance, the `laws_initiatives` spider accepts an additional `llp` parameter:

```
python manage.py scrape crawl -a llp=21 laws_initiatives
```

In the spider itself, this parameter can be extracted like this:

```
def __init__(self, **kw):
    super(LawsInitiativesSpider, self).__init__(**kw)
    if 'llp' in kw:
        try:
            self.LLP = [int(kw['llp'])]
        except:
            pass
    # [...]
```

Extractors

Extractors take over the heavy lifting - by translating the raw html source code they are handed into organized data, ready for insertion into the database.

The simplest extractor just inherits from `parlament.resources.extractors.SingleExtractor`, which provides an `xt` method and utilizes a simple class variable containing the XPath expression to extract, expecting it to evaluate to exactly one result. For instance, the `title` of a `law` detail page might be extracted by the following class:

```
from parlament.resources.extractors import SingleExtractor

class LAW:
    class TITLE(SingleExtractor):
        XPATH = '//*[@id="inhalt"]/text()'
```

Similarly, to simply extract a list of items based on an XPath expression, the following code could be used:

```
class LAW:
    class KEYWORDS(MultiExtractor):
        XPATH = '//*[@id="schlagwortBox"]/ul//li/a/text()'
```

In reality, many of the extractors overwrite the `xt` method to implement more complex extractions.

Search Provider: Haystack, Elasticsearch

This section describes the search and indexing implementation.

Basics

The current application relies on [Django Haystack](#), a high-level framework brokering between Django and a search backend. This search backend is currently [ElasticSearch](#), but could be interchanged for [Apache SOLR](#), should the need arise.

Re-Indexing

For now, reindexing (or updating the index, for that matter), is only done manually. To have all data indexed, just run:

```
python manage.py rebuild_index
```

for a full rebuild (wipes the indices first), or:

```
python manage.py update_index
```

to perform a simple update. For this to succeed, make sure ElasticSearch is up and running.

SearchViews

Searching is split between different contexts, represented by different Django views (cf. `offenesparlament/search_views.py`):

1. Main Search (all indices), at `/search`
2. Persons, at `personen/search`
3. Laws, at `gesetze/search`
4. debates, at `debatten/search`

Each view determines the available *facets* - for instance, the *Person* view returns, among others, faceting information for the person's *party* in it's results.

The views all inherit from `JsonSearchView`, an adaptation of Haystack's `SearchView` that, instead of rendering a template, returns JSON data to be processed by the frontend.

Each accepts a query parameter, *q*, and a list of *facet filters*, named like the facets available for that view:

Main Search

- No Facets

Persons

- party: A person's party, for instance, SPÖ
- birthplace: A persons birthplace
- deathplace: A persons deathplace
- occupation: A persons occupation
- lps: The legislative period(s) a person was/is active during
- ts: The timestamp that entry was last updaten (from the parlament site)

Laws

- category: A law's category
- keywords: A law's assigned keywords
- lrp: The legislative period of a law
- ts: The timestamp that entry was last updaten (from the parlament site)

Debates

- `llp`: The legislative period a debate fell into
- `debate_type`: either NR or BR (Nationalrat/Bundesrat)
- `date`: The date the debate happened

Each of the facet filters means each resulting entry must *contain* the term, but it is not specifying exact searches; for instance, filtering fields that might contain multiple entries like a person's active legislative periods, for instance, will return all persons that have the period in question *in* their list, not just persons whose list contains *only* the period in question.

The query parameter searches in the index's `text` field - an aggregate field containing most of the other fields to allow more specific searches.

All parameters have to be supplied as GET-Parameters. A typical request might look like this:

```
http://offenesparlament.vm:8000/personen/search?q=Franz&llps=XXIV&party=SP%C3%96
```

and would return the following JSON data:

```
{
  "facets": {
    "fields": {
      "party": [
        [
          "SP\u00d6",
          2
        ]
      ],
      "birthplace": [
        [
          " Wien",
          1
        ],
        [
          " Wels",
          1
        ]
      ],
      "llps": [
        [
          "XXIV",
          2
        ],
        [
          "XXIII",
          2
        ],
        [
          "XXV",
          1
        ],
        [
          "XXII",
          1
        ],
        [
          "XXI",
          1
        ],
        [
          "XX",
          1
        ]
      ]
    }
  }
}
```

```

    ]
  ],
  "deathplace": [
    [
      "",
      2
    ]
  ],
  "occupation": [
    [
      " Kaufmann",
      1
    ],
    [
      " Elektromechaniker",
      1
    ]
  ]
},
"dates": {
},
"queries": {
}
},
"result": [
  {
    "birthplace": " Wien",
    "party_exact": "SP\u00d6",
    "llps_exact": [
      "XXIV",
      "XXIII",
      "XXII",
      "XXI",
      "XX"
    ],
    "text": "PAD_03599\nFranz Riepl\nRiepl Franz\n Wien\n\n Elektromechaniker",
    "birthdate": "1949-03-23T00:00:00",
    "llps": [
      "XXIV",
      "XXIII",
      "XXII",
      "XXI",
      "XX"
    ],
    "deathdate": null,
    "deathplace": "",
    "full_name": "Franz Riepl",
    "occupation_exact": " Elektromechaniker",
    "party": "SP\u00d6",
    "deathplace_exact": "",
    "birthplace_exact": " Wien",
    "reversed_name": "Riepl Franz",
    "source_link": "http://www.parlament.gv.at/WWER/PAD_03599/index.shtml",
    "occupation": " Elektromechaniker"
  },
  {
    "birthplace": " Wels",
    "party_exact": "SP\u00d6",

```

```

    "llps_exact": [
        "XXIV",
        "XXIII",
        "XXV"
    ],
    "text": "PAD_35495\nFranz Kirchgatterer\nKirchgatterer Franz\n Wels\n\n Kaufmann",
    "birthdate": "1953-09-24T00:00:00",
    "llps": [
        "XXIV",
        "XXIII",
        "XXV"
    ],
    "deathdate": null,
    "deathplace": "",
    "full_name": "Franz Kirchgatterer",
    "occupation_exact": " Kaufmann",
    "party": "SP\u00d6",
    "deathplace_exact": "",
    "birthplace_exact": " Wels",
    "reversed_name": "Kirchgatterer Franz",
    "source_link": "http://www.parlament.gv.at/WWER/PAD_35495/index.shtml",
    "occupation": " Kaufmann"
}
]
}

```

Facet Only Search

Besides the normal, query-based search, it is possible to retrieve only the facets (for an empty query, for instance). This is necessary to allow filling of dropdown/selection boxes before the first search. A typical request might then look like this:

```
http://offenesparlament.vm:8000/personen/search?q=&only_facets=true
```

But this facet-only search also works with a query, should that be necessary:

```
http://offenesparlament.vm:8000/personen/search?q=Mayer&only_facets=true
```

The result looks like the above-mentioned search result, but always contains an empty list in the 'results' field.

Paging

In addition to the query arguments for filtering and faceting, the search views also automatically limit the results to allow for smooth paging. Two parameters govern this behaviour: *offset* and *limit*.

Offset returns search results from the given integer on - so, for a search that produced 100 results, an offset value of '20' would only return results 20 to 100. If no *offset* value is given, the view assumes '0' and returns results starting with the first one.

Limit restricts the amount of results per page; with the abovementioned example and a *limit* value of '50', the query would only return results 20 through 70. If no *limit* is given, the view assumes a default of 50 results. This can be changed in the `offenesparlament/constants.py` file.

Fieldsets

Given the amount of data in the index (particularly the debate statements), returning the entirety of an object including all of its fields is not performant enough for long lists of results. To combat that issue, the concept of predefined fieldsets has been introduced. Each index class now contains a FIELDSET dictionary which defines the available fieldsets. The debate class, for instance, contains the following fieldsets:

```
FIELDSETS = {
    'all': ['text', 'date', 'title', 'debate_type', 'protocol_url', 'detail_url', 'nr', 'l
```

```

    'list': ['text', 'date', 'title', 'debate_type', 'protocol_url', 'detail_url', 'nr', '']
  }

```

The dictionary key describes the fieldset, and the value consists of a list of all fields that should be returned when requesting that fieldset.

Per default, the search view only returns the 'list' fieldset; if a search request must return all available data, the 'fieldset' parameter allows querying the for a specific fieldset:

```

http://offenesparlament.vm:8000/personen/search?parl_id=PAD_65677&fieldset=all

```

Single Result Search

The normal search will always be aimed at finding a list of entries that match the query, but for the detail pages (and the subscriptions), a search mode that returns exactly one result was needed. This view can be accessed by, instead of adding in facet filters, filter by 'parl_id' (for persons) or by 'parl_id' and 'llp' for laws.

For instance, a typical request for a single person's data might look like this:

```

http://offenesparlament.vm:8000/personen/search?parl_id=PAD_65677&detail=True

```

This would yield a result similar to the following:

```

{
  "result": [
    {
      "debate_statements": "[]",
      "birthplace": " Kufstein (Tirol)",
      "deathdate": null,
      "statements": [
        {
          "law_id": 6814,
          "law_slug": "/gesetze/XXV/905_d.B./",
          "law_category": "Regierungsvorlage: Bundes (verfassung)",
          "date": "2016-02-11",
          "protocol_url": null,
          "law": "Pflanzenschutzgesetz 2011, \u00c4nderung",
          "type": "Pro"
        },
        {
          "law_id": 7298,
          "law_slug": "/gesetze/XXV/81-A/",
          "law_category": "Selbst\u00e4ndiger Antrag",
          "date": "2014-01-31",
          "protocol_url": "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00826/SEITE_0026.html",
          "law": "Bundesministeriengesetz 1986, \u00c4nderung",
          "type": "Contra"
        },
        {
          "law_id": 7285,
          "law_slug": "/gesetze/XXV/111-A/",
          "law_category": "Selbst\u00e4ndiger Antrag",
          "date": "2014-02-26",
          "protocol_url": "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00827/SEITE_0122.html",
          "law": "Umweltvert\u00e4glichkeitspr\u00e4fungsgesetz 2000, \u00c4nderung",
          "type": "Contra"
        },
        {
          "law_id": 7268,
          "law_slug": "/gesetze/XXV/14_d.B./",

```

```

        \ "law_category\ ": \ "Regierungsvorlage: Bundes( verfassung
        \ "date\ ": \ "2014-04-10\ ",
        \ "protocol_url\ ": \ "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00828/SEITE_0181.html\ ",
        \ "law\ ": \ "26. StVO-Novelle\ ",
        \ "type\ ": \ "Pro\ "
    },
    {
        \ "law_id\ ": 7262,
        \ "law_slug\ ": \ "/gesetze/XXV/113-A/\ ",
        \ "law_category\ ": \ "Selbst\ \u00e4ndiger Antrag\ ",
        \ "date\ ": \ "2014-04-10\ ",
        \ "protocol_url\ ": \ "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00828/SEITE_0185.html\ ",
        \ "law\ ": \ "Kraftfahrgesetz 1967,
        \ \u00c4nderung\ ",
        \ "type\ ": \ "Pro\ "
    },
    {
        \ "law_id\ ": 7259,
        \ "law_slug\ ": \ "/gesetze/XXV/261-A/\ ",
        \ "law_category\ ": \ "Selbst\ \u00e4ndiger Antrag\ ",
        \ "date\ ": \ "2014-04-10\ ",
        \ "protocol_url\ ": \ "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00828/SEITE_0187.html\ ",
        \ "law\ ": \ "SP-V-Gesetz-Novelle 2014\ ",
        \ "type\ ": \ "Contra\ "
    },
    {
        \ "law_id\ ": 7253,
        \ "law_slug\ ": \ "/gesetze/XXV/5_d.B./\ ",
        \ "law_category\ ": \ "Regierungsvorlage: Staatsvertrag\ ",
        \ "date\ ": \ "2014-05-15\ ",
        \ "protocol_url\ ": \ "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00829/SEITE_0128.html\ ",
        \ "law\ ": \ "Internes Abkommen \ \u00fcr die Finanzierung der vorgesehenen H
        \ "type\ ": \ "Pro\ "
    },
    {
        \ "law_id\ ": 7252,
        \ "law_slug\ ": \ "/gesetze/XXV/13_d.B./\ ",
        \ "law_category\ ": \ "Regierungsvorlage: Staatsvertrag\ ",
        \ "date\ ": \ "2014-05-15\ ",
        \ "protocol_url\ ": \ "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00829/SEITE_0128.html\ ",
        \ "law\ ": \ "Abkommen mit der Internationalen Organisation f\ \u00fcr Migratio
        \ "type\ ": \ "Pro\ "
    },
    {
        \ "law_id\ ": 7251,
        \ "law_slug\ ": \ "/gesetze/XXV/15_d.B./\ ",
        \ "law_category\ ": \ "Regierungsvorlage: Staatsvertrag\ ",
        \ "date\ ": \ "2014-05-15\ ",
        \ "protocol_url\ ": \ "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00829/SEITE_0128.html\ ",
        \ "law\ ": \ "Abkommen mit Zypern \ \u00fcr die Verwendung von Flugh\ \u00e4f
        \ "type\ ": \ "Pro\ "
    },
    {
        \ "law_id\ ": 7247,
        \ "law_slug\ ": \ "/gesetze/XXV/29_d.B./\ ",
        \ "law_category\ ": \ "Regierungsvorlage: Staatsvertrag\ ",
        \ "date\ ": \ "2014-05-15\ ",
        \ "protocol_url\ ": \ "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00829/SEITE_0128.html\ ",
        \ "law\ ": \ "Rahmenabkommen \ \u00fcr Partnerschaft\ ",
        \ "type\ ": \ "Pro\ "
    }

```

```

    },
    {
        \"law_id\":7245,
        \"law_slug\":\"/gesetze/XXV/71_d.B./\",
        \"law_category\":          \"Regierungsvorlage:Staatsvertrag\",
        \"date\": \"2014-05-15\",
        \"protocol_url\":\"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00829/SEITE_0128.html\",
        \"law\": \"Abkommen zur zweiten \u00c4nderung des Partnerschaftsabkommens
Karibik- und Pazifik-Staaten und der EU samt Schlussakte einschlie\u00dfli
        \"type\": \"Pro\"
    },
    {
        \"law_id\":7238,
        \"law_slug\":\"/gesetze/XXV/53_d.B./\",
        \"law_category\":          \"Regierungsvorlage:Bundes(verfassung
        \"date\": \"2014-05-28\",
        \"protocol_url\":\"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00830/SEITE_0061.html\",
        \"law\": \"Budgetbegleitgesetz 2014\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\":7235,
        \"law_slug\":\"/gesetze/XXV/101_d.B./\",
        \"law_category\":          \"Regierungsvorlage:Bundes(verfassung
        \"date\": \"2014-05-28\",
        \"protocol_url\":\"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00830/SEITE_0061.html\",
        \"law\": \"Grunderwerbsteuergesetz 1987,
\u00c4nderung\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\":7234,
        \"law_slug\":\"/gesetze/XXV/126_d.B./\",
        \"law_category\":          \"Regierungsvorlage:Bundes(verfassung
        \"date\": \"2014-05-28\",
        \"protocol_url\":\"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00830/SEITE_0061.html\",
        \"law\": \"Begr\u00fcndung von Vorbelastungen durch die Bundesministerin f
Innovation und Technologie\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\":7218,
        \"law_slug\":\"/gesetze/XXV/142_d.B./\",
        \"law_category\":          \"Regierungsvorlage:Bundes(verfassung
        \"date\": \"2014-06-26\",
        \"protocol_url\":\"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00831/SEITE_0043.html\",
        \"law\": \"Marktordnungsgesetz 2007,
\u00c4nderung\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\":7143,
        \"law_slug\":\"/gesetze/XXV/318_d.B./\",
        \"law_category\":          \"Regierungsvorlage:Bundes(verfassung
        \"date\": \"2014-12-04\",
        \"protocol_url\":\"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00836/SEITE_0060.html\",
        \"law\": \"Eisenbahngesetz 1957,
Unfalluntersuchungsgesetz,
\u00c4nderung\",
        \"type\": \"Contra\"
    }

```



```

    },
    {
        \"law_id\":7137,
        \"law_slug\":\"/gesetze/XXV/295-A/\",
        \"law_category\": \"Selbst\\u00e4ndiger Antrag\",
        \"date\": \"2014-12-04\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00836/SEITE_0065.html\",
        \"law\": \"Stra\\u00dfenverkehrsordnung 1960,
        \\u00c4nderung\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\":7135,
        \"law_slug\":\"/gesetze/XXV/721-A/\",
        \"law_category\": \"Selbst\\u00e4ndiger Antrag\",
        \"date\": \"2014-12-04\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00836/SEITE_0065.html\",
        \"law\": \"Kraftfahrgesetz 1967,
        \\u00c4nderung\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\":7100,
        \"law_slug\":\"/gesetze/XXV/364_d.B./\",
        \"law_category\": \"Regierungsvorlage:Bundes(verfassung)
        \\u00c4nderung\",
        \"date\": \"2014-12-18\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00837/SEITE_0043.html\",
        \"law\": \"Wehrgesetz 2001,
        \\u00c4nderung\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\":7096,
        \"law_slug\":\"/gesetze/XXV/369_d.B./\",
        \"law_category\": \"Regierungsvorlage:Bundes(verfassung)
        \\u00c4nderung\",
        \"date\": \"2014-12-18\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00837/SEITE_0185.html\",
        \"law\": \"Universit\\u00e4tsgesetz 2002,
        Hochschulgesetz 2005,
        \\u00c4nderung\",
        \"type\": \"Pro\"
    },
    {
        \"law_id\":7094,
        \"law_slug\":\"/gesetze/XXV/371_d.B./\",
        \"law_category\": \"Regierungsvorlage:Bundes(verfassung)
        \\u00c4nderung\",
        \"date\": \"2014-12-18\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00837/SEITE_0078.html\",
        \"law\": \"Chemikaliengesetz 1996,
        Bundeskriminalamt-Gesetz,
        \\u00c4nderung\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\":7080,
        \"law_slug\":\"/gesetze/XXV/445_d.B./\",
        \"law_category\": \"Regierungsvorlage:Bundes(verfassung)
        \\u00c4nderungsgesetz 2015\",
        \"date\": \"2015-02-05\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00838/SEITE_0078.html\",
        \"law\": \"Fortpflanzungsmedizinrechts-\\u00c4nderungsgesetz 2015\",
    }

```

```

    \ "type\":\ "Pro\"
  },
  {
    \ "law_id\":7032,
    \ "law_slug\":\ "/gesetze/XXV/460_d.B./\",
    \ "law_category\":\ "Regierungsvorlage:Bundes(verfassung)",
    \ "date\":\ "2015-05-07\",
    \ "protocol_url\":\ "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00841/SEITE_0107.html\",
    \ "law\":\ "Passagier- und Fahrgastrechteagenturgesetz\",
    \ "type\":\ "Contra\"
  },
  {
    \ "law_id\":7030,
    \ "law_slug\":\ "/gesetze/XXV/510_d.B./\",
    \ "law_category\":\ "Regierungsvorlage:Bundes(verfassung)",
    \ "date\":\ "2015-05-07\",
    \ "protocol_url\":\ "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00841/SEITE_0098.html\",
    \ "law\":\ "Kraftfahrliienigesetz,
    \u00c4nderung\",
    \ "type\":\ "Contra\"
  },
  {
    \ "law_id\":7029,
    \ "law_slug\":\ "/gesetze/XXV/511_d.B./\",
    \ "law_category\":\ "Regierungsvorlage:Bundes(verfassung)",
    \ "date\":\ "2015-05-07\",
    \ "protocol_url\":\ "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00841/SEITE_0098.html\",
    \ "law\":\ "\u00d6ffentlicher Personennah- und Regionalverkehrsgesetz 1999,
    \u00c4nderung\",
    \ "type\":\ "Contra\"
  },
  {
    \ "law_id\":7012,
    \ "law_slug\":\ "/gesetze/XXV/584_d.B./\",
    \ "law_category\":\ "Regierungsvorlage:Bundes(verfassung)",
    \ "date\":\ "2015-06-03\",
    \ "protocol_url\":\ "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00842/SEITE_0083.html\",
    \ "law\":\ "Bundesbahngesetz,
    \u00c4nderung\",
    \ "type\":\ "Contra\"
  },
  {
    \ "law_id\":7007,
    \ "law_slug\":\ "/gesetze/XXV/585_d.B./\",
    \ "law_category\":\ "Regierungsvorlage:Bundes(verfassung)",
    \ "date\":\ "2015-06-03\",
    \ "protocol_url\":\ "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00842/SEITE_0083.html\",
    \ "law\":\ "Dienstrechts-Novelle 2015\",
    \ "type\":\ "Contra\"
  },
  {
    \ "law_id\":6978,
    \ "law_slug\":\ "/gesetze/XXV/631_d.B./\",
    \ "law_category\":\ "Regierungsvorlage:Bundes(verfassung)",
    \ "date\":\ "2015-07-02\",
    \ "protocol_url\":\ "/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00843/SEITE_0123.html\",
    \ "law\":\ "F\u00fchrerscheingesetz (16. FSG-Novelle)\",
    \ "type\":\ "Pro\"
  },
  {

```

```

        \"law_id\":6977,
        \"law_slug\":\"/gesetze/XXV/1185-A/\",
        \"law_category\":\"Selbst\\u00e4ndiger Antrag\",
        \"date\":\"2015-07-02\",
        \"protocol_url\":\"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00843/SEITE_0117.html\",
        \"law\":\"Kraftfahrgesetz 1967; \\u00c4nderung\",
        \"type\":\"Pro\"
    },
    {
        \"law_id\":6976,
        \"law_slug\":\"/gesetze/XXV/1191-A/\",
        \"law_category\":\"Selbst\\u00e4ndiger Antrag\",
        \"date\":\"2015-07-02\",
        \"protocol_url\":\"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00843/SEITE_0117.html\",
        \"law\":\"Kraftfahrgesetz 1967; \\u00c4nderung\",
        \"type\":\"Pro\"
    },
    {
        \"law_id\":6959,
        \"law_slug\":\"/gesetze/XXV/501_d.B./\",
        \"law_category\":                \"Regierungsvorlage:Staatsvertrag\",
        \"date\":\"2015-07-23\",
        \"protocol_url\":null,
        \"law\":\"Erkl\\u00e4rung \\u00fcb er die Zur\\u00fckziehung der \\u00f6st
        15                und 17 sowie der Erkl\\u00e4rungen zu Art. 38 des \\
        \"type\":\"Pro\"
    },
    {
        \"law_id\":6957,
        \"law_slug\":\"/gesetze/XXV/530_d.B./\",
        \"law_category\":                \"Regierungsvorlage:Bundes(verfassung
        \"date\":\"2015-07-23\",
        \"protocol_url\":null,
        \"law\":\"Gentechnikgesetz,
        \\u00c4nderung\",
        \"type\":\"Pro\"
    },
    {
        \"law_id\":6952,
        \"law_slug\":\"/gesetze/XXV/617_d.B./\",
        \"law_category\":                \"Regierungsvorlage:Bundes(verfassung
        \"date\":\"2015-07-23\",
        \"protocol_url\":null,
        \"law\":\"EZG-Novelle 2015\",
        \"type\":\"Contra\"
    },
    {
        \"law_id\":6944,
        \"law_slug\":\"/gesetze/XXV/673_d.B./\",
        \"law_category\":                \"Regierungsvorlage:Bundes(verfassung
        \"date\":\"2015-07-23\",
        \"protocol_url\":null,
        \"law\":\"Gentechnik-Anbauverbots-Rahmengesetz; Sortenschutzgesetz,
        \\u00c4nderung\",
        \"type\":\"Pro\"
    },
    {
        \"law_id\":6940,
        \"law_slug\":\"/gesetze/XXV/680_d.B./\",
        \"law_category\":                \"Regierungsvorlage:Bundes(verfassung

```

```

        \"date\": \"2015-07-23\",
        \"protocol_url\": null,
        \"law\": \"Marktordnungsgesetz 2007,
        \\u00c4nderung\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 6932,
        \"law_slug\": \"/gesetze/XXV/688_d.B./\",
        \"law_category\": \"Regierungsvorlage: Bundes (verfassung)
        \"date\": \"2015-07-23\",
        \"protocol_url\": null,
        \"law\": \"Erbrechts-\\u00c4nderungsgesetz 2015\",
        \"type\": \"Pro\"
    },
    {
        \"law_id\": 6931,
        \"law_slug\": \"/gesetze/XXV/689_d.B./\",
        \"law_category\": \"Regierungsvorlage: Bundes (verfassung)
        \"date\": \"2015-07-23\",
        \"protocol_url\": null,
        \"law\": \"Strafrechts-\\u00e4nderungsgesetz 2015\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 6928,
        \"law_slug\": \"/gesetze/XXV/693_d.B./\",
        \"law_category\": \"Regierungsvorlage: Staatsvertrag\",
        \"date\": \"2015-07-23\",
        \"protocol_url\": null,
        \"law\": \"In Doha beschlossene \\u00c4nderung des Protokolls von Kyoto zum
        \"type\": \"Pro\"
    },
    {
        \"law_id\": 6927,
        \"law_slug\": \"/gesetze/XXV/694_d.B./\",
        \"law_category\": \"Regierungsvorlage: Staatsvertrag\",
        \"date\": \"2015-07-23\",
        \"protocol_url\": null,
        \"law\": \"Vereinbarung zwischen der Europ\\u00e4ischen Union und ihren Mit
        \"type\": \"Pro\"
    },
    {
        \"law_id\": 6925,
        \"law_slug\": \"/gesetze/XXV/696_d.B./\",
        \"law_category\": \"Regierungsvorlage: Bundes (verfassung)
        \"date\": \"2015-07-23\",
        \"protocol_url\": null,
        \"law\": \"Umweltinformationsgesetz,
        \\u00c4nderung\",
        \"type\": \"Pro\"
    },
    {
        \"law_id\": 6921,
        \"law_slug\": \"/gesetze/XXV/1181-A/\",
        \"law_category\": \"Selbst\\u00e4ndiger Antrag\",
        \"date\": \"2015-07-23\",
        \"protocol_url\": null,
        \"law\": \"Forstgesetz 1975; \\u00c4nderung\",
        \"type\": \"Pro\"
    }

```

```

    },
    {
        \"law_id\":6912,
        \"law_slug\":\"/gesetze/XXV/775_d.B./\",
        \"law_category\": \"Regierungsvorlage:Bundes(verfassung)\",
        \"date\": \"2015-09-25\",
        \"protocol_url\":null,
        \"law\": \"27. StVO-Novelle\",
        \"type\": \"Pro\"
    },
    {
        \"law_id\":6911,
        \"law_slug\":\"/gesetze/XXV/1295-A/\",
        \"law_category\": \"Selbst\\u00e4ndiger Antrag\",
        \"date\": \"2015-09-25\",
        \"protocol_url\":null,
        \"law\": \"Unterbringung und Aufteilung von hilfs- und schutzbed\\u00fcrfti\",
        \"type\": \"Pro\"
    },
    {
        \"law_id\":6910,
        \"law_slug\":\"/gesetze/XXV/1296-A/\",
        \"law_category\": \"Selbst\\u00e4ndiger Antrag\",
        \"date\": \"2015-09-25\",
        \"protocol_url\":null,
        \"law\": \"Fremdenpolizeigesetz 2005; \\u00c4nderung\",
        \"type\": \"Pro\"
    },
    {
        \"law_id\":6887,
        \"law_slug\":\"/gesetze/XXV/799_d.B./\",
        \"law_category\": \"Regierungsvorlage:Staatsvertrag\",
        \"date\": \"2015-10-29\",
        \"protocol_url\":null,
        \"law\": \"halbt\\u00e4gig kostenlose und verpflichtende fr\\u00fche F\\u00\",
        \"type\": \"Pro\"
    },
    {
        \"law_id\":6872,
        \"law_slug\":\"/gesetze/XXV/823_d.B./\",
        \"law_category\": \"Regierungsvorlage:Bundes(verfassung)\",
        \"date\": \"2015-11-19\",
        \"protocol_url\":null,
        \"law\": \"Strahlenschutzgesetz, \\u00c4nderung\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\":6858,
        \"law_slug\":\"/gesetze/XXV/821_d.B./\",
        \"law_category\": \"Regierungsvorlage:Bundes(verfassung)\",
        \"date\": \"2015-12-03\",
        \"protocol_url\":null,
        \"law\": \"Budgetbegleitgesetz 2016\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\":6857,
        \"law_slug\":\"/gesetze/XXV/846_d.B./\",
        \"law_category\": \"Regierungsvorlage:Bundes(verfassung)\",

```

```

        \"date\": \"2015-12-03\",
        \"protocol_url\": null,
        \"law\": \"Begr\u00fcdung von Vorbelastungen durch den Bundesminister f\u00fcr
        Innovation und Technologie\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 6856,
        \"law_slug\": \"/gesetze/XXV/883_d.B./\",
        \"law_category\": \"Bericht und Antrag\",
        \"date\": \"2015-12-03\",
        \"protocol_url\": null,
        \"law\": \"Rechnungshofgesetz 1948,
        \u00c4nderung\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 5778,
        \"law_slug\": \"/gesetze/XXIV/2201_d.B./\",
        \"law_category\": \"Regierungsvorlage:Staatsvertrag\",
        \"date\": \"2013-06-06\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00821/SEITE_0126.html\",
        \"law\": \"\u00dcbereinkommen \u00fcbcr das Europ\u00e4ische Forstinstitut\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 5776,
        \"law_slug\": \"/gesetze/XXIV/2252_d.B./\",
        \"law_category\": \"Regierungsvorlage:Bundes(verfassung)sgesetz\",
        \"date\": \"2013-06-06\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00821/SEITE_0114.html\",
        \"law\": \"Umweltvertr\u00e4glichkeitspr\u00fcdungsgesetz 2000,
        \u00c4nderung; Bundesgesetz \u00fcbcr den Umweltsenat,
        Aufhebung\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 5773,
        \"law_slug\": \"/gesetze/XXIV/2290_d.B./\",
        \"law_category\": \"Regierungsvorlage:Bundes(verfassung)sgesetz\",
        \"date\": \"2013-06-06\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00821/SEITE_0114.html\",
        \"law\": \"Verwaltungsgerichtsbarkeits-Anpassungsgesetz \u00e4 2013 Umwelt,
        Abfall,
        Wasser\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 5772,
        \"law_slug\": \"/gesetze/XXIV/2291_d.B./\",
        \"law_category\": \"Regierungsvorlage:Bundes(verfassung)sgesetz\",
        \"date\": \"2013-06-06\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00821/SEITE_0126.html\",
        \"law\": \"Verwaltungsgerichtsbarkeits-Anpassungsgesetz-BMLFUW \u00e4 2013 Land\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 5771,
        \"law_slug\": \"/gesetze/XXIV/2292_d.B./\",
        \"law_category\": \"Regierungsvorlage:Bundes(verfassung)sgesetz\",

```

```

        \"date\": \"2013-06-06\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00821/SEITE_0114.html\",
        \"law\": \"Umweltrechtsanpassungsgesetz 2013\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 5770,
        \"law_slug\": \"/gesetze/XXIV/2293_d.B./\",
        \"law_category\": \"Regierungsvorlage: Bundes (verfassungsgesetz)\",
        \"date\": \"2013-06-06\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00821/SEITE_0114.html\",
        \"law\": \"AWG-Novelle Industrieemissionen,
        Altlastensanierungsgesetz,
        \u00c4nderung\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 5768,
        \"law_slug\": \"/gesetze/XXIV/2295_d.B./\",
        \"law_category\": \"Regierungsvorlage: Bundes (verfassungsgesetz)\",
        \"date\": \"2013-06-06\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00821/SEITE_0105.html\",
        \"law\": \"Klimaschutzgesetz,
        \u00c4nderung\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 5766,
        \"law_slug\": \"/gesetze/XXIV/2297_d.B./\",
        \"law_category\": \"Regierungsvorlage: Bundes (verfassungsgesetz)\",
        \"date\": \"2013-06-06\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00821/SEITE_0126.html\",
        \"law\": \"Agrarrechts\u00e4nderungsgesetz 2013\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 5746,
        \"law_slug\": \"/gesetze/XXIV/2321_d.B./\",
        \"law_category\": \"Regierungsvorlage: Bundes (verfassungsgesetz)\",
        \"date\": \"2013-06-26\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00822/SEITE_0204.html\",
        \"law\": \"Emissionsschutzgesetz f\u00fcr Kesselanlagen\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 5741,
        \"law_slug\": \"/gesetze/XXIV/2337_d.B./\",
        \"law_category\": \"Regierungsvorlage: Bundes (verfassungsgesetz)\",
        \"date\": \"2013-06-26\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00822/SEITE_0204.html\",
        \"law\": \"Gewerbeordnung 1994,
        \u00c4nderung\",
        \"type\": \"Contra\"
    },
    {
        \"law_id\": 5728,
        \"law_slug\": \"/gesetze/XXIV/2316-A/\",
        \"law_category\": \"Selbst\u00e4ndiger Antrag\",
        \"date\": \"2013-06-26\",
        \"protocol_url\": \"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00822/SEITE_0174.html\",
    }

```

```

        \ "law\":\"Nachhaltigkeit,
        den Tierschutz,
        den umfassenden Umweltschutz,
        die Sicherstellung der Wasser- und Lebensmittelversorgung und die Forschun
        \ "type\":\"Contra\"
    },
    {
        \ "law_id\":5691,
        \ "law_slug\":\"/gesetze/XXIV/2361_d.B./\",
        \ "law_category\":          \"Regierungsvorlage:Staatsvertrag\",
        \ "date\":\"2013-07-18\",
        \ "protocol_url\":\"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00823/SEITE_0087.html\",
        \ "law\":\"2. Vereinbarung gem\\u00e4\\u00df Artikel 15a B-VG \\u00fber Vo
        \ "type\":\"Pro\"
    },
    {
        \ "law_id\":5671,
        \ "law_slug\":\"/gesetze/XXIV/2408_d.B./\",
        \ "law_category\":          \"Regierungsvorlage:Bundes(verfassung)
        \ "date\":\"2013-07-18\",
        \ "protocol_url\":\"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00823/SEITE_0201.html\",
        \ "law\":\"AWG-Novelle Verpackung,
        \\u00c4nderung\",
        \ "type\":\"Contra\"
    },
    {
        \ "law_id\":5651,
        \ "law_slug\":\"/gesetze/XXIV/2447_d.B./\",
        \ "law_category\":          \"Regierungsvorlage:Staatsvertrag\",
        \ "date\":\"2013-07-18\",
        \ "protocol_url\":\"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00823/SEITE_0239.html\",
        \ "law\":\"\\u00dcbereinkommen \\u00fber ein Einheitliches Patentgericht\"
        \ "type\":\"Contra\"
    },
    {
        \ "law_id\":5644,
        \ "law_slug\":\"/gesetze/XXIV/2323-A/\",
        \ "law_category\":\"Selbst\\u00e4ndiger Antrag\",
        \ "date\":\"2013-07-18\",
        \ "protocol_url\":\"/PAKT/VHG/BR/BR/SITZ/BR/SITZ_00823/SEITE_0117.html\",
        \ "law\":\"Elektrizit\\u00e4tswirtschafts- und -organisationsgesetz 2010,
        Gaswirtschaftsgesetz 2011,
        Energie-Control-Gesetz,
        \\u00c4nderung\",
        \ "type\":\"Pro\"
    }
}
]
"
    text\":\"PAD_65677\\nMag. Nicole Schreyer\\nSchreyer Nicole, Mag.\\n Kufstein (Tirol
    \"mandates\": \"[{\ \"function\": {\ \"short\": \"\", \ \"title\": \"Abgeordnete(r) zum Bu
    \"ts\": \"2016-02-12T02:15:00\",
    \"birthdate\": \"1977-02-05T00:00:00\",
    \"reversed_name\": \"Schreyer Nicole, Mag.\",
    \"photo_copyright\": \"© Parlamentsdirektion / Photo Simonis\",
    \"llps_numeric\": [
        25,
        24
    ],
    \"deathplace\": \"\",
    \"full_name\": \"Mag. Nicole Schreyer\",
    \"source_link\": \"http://www.parlament.gv.at/WWER/PAD_65677/index.shtml\",

```



```

        "party": "Grüne",
        "llps": ["'aktuell seit 2013-10-29 (XXV)', '2008-10-28 - 2013-10-28 (XXIV)']",
        "internal_link": "/personen/PAD_65677/Mag-Nicole-Schreyer/",
        "photo_link": "http://www.parlament.gv.at/WWER/PAD_65677/6020956_180.jpg",
        "parl_id": "PAD_65677",
        "occupation": " Biologin"
    }
]
}

```

Indices

WARNING: Currently, only three separate indices exist, one for the Laws, one for the Persons and one for the Debates. These are subject to heavy development in the future and will change a lot still, so this documentation will remain mostly blank for now.

The indices are defined in `op_scraper/search_indexes.py`. Each index contains a `text` field, which aggregates the objects' data into a single, text-based field, which Haystack uses as the default search field. The exact makeup of this field is defined in `templates`, located at `offenesparlament/templates/search/indexes/op_scraper/*_text.html`.

Django Admin Interface

The Django Administration Interface has been extended with a few vital functions to make maintenance easier.

Manually trigger scraping

Besides the usual CRUD-Interfaces, which should only be used for debugging purposes, given that the site's entire data should be automatically scraped from the parliament website, a new block called *Scraping Management* has been added, which allows manually triggering one of the following spiders:

1. Legislative Periods
2. Persons
3. Administrations
4. Pre-Laws
5. Laws

The order of the scrapers above represents their dependencies on each other; for instance, scanning laws includes votes and speeches by Persons, and relies on the Person in question having been scraped before. To be sure, the order above should be maintained in all scraping processes.

Behind the scenes, the scraping view `offenesparlament.op_scraper.admin_views.trigger_scrape` calls the celery task `offenesparlament.op_scraper.tasks.scrape`, which in turn finds and executes the requested scraper, but wraps this in a `django-reversion` block to create new revisions of all affected database objects.

Import-Export

To allow import-export, the `import_export` django module is being used, overwriting the `templates/admin/changelist.html` admin template to be compatible with `django_reversion`.

Frontend

This section describes the frontend code and how to build it

Tech

- [React](#)
- [VisualSearch](#)

Building

We use grunt to build our frontend code. It is generated from the sources in `client/` and is output to `offenesparlament/offenesparlament/static/`.

We commit all generated files to git.

Grunt is already installed in the vagrant VM, so you should be able to run the build task from the VM right away in the dir `/vagrant`:

```
grunt dev
```

This task watches all source files and rebuilds if necessary.

To use BrowserSync and have the browser reload every time frontend files change, run:

```
grunt reloading
```

Grunt too slow?

If grunt is too slow running inside the VM (Probably due to file-watching on the host system) you'll have to install the following on your computer:

- [Node.js and npm](#) (Make sure you run recent versions of both)
- [grunt-CLI](#)
- [Sass](#) (You might need to install Ruby first)

Then you can run grunt tasks on your computer from the project dir `OffenesParlament` (where the `Gruntfile.coffee` is located)

Subscriptions via Email

This section describes the subscription structure and implementation.

Basics

An important feature of OffenesParlament.at is the ability to subscribe to certain pages for email-updates, if and when they get changes through a scrape. This includes search views as well as detail views. To maintain a maximum of privacy for the user while also keeping the user's data and email-adresses safe and spam-free, a verification process is used that provides one-time links for verifying the subscriptions as well as viewing a list of subscriptions for a given user. The following functionalities have been implemented:

- Subscription of a page via web form
- Verification email sending, Verification view
- Subscription List view
- Subscription List URL reminder email sending
- Subscription Deletion

The following things are in beta or have yet to be implemented:

- Frontend Design for views
- Actual "Changes" email based on search results
- Frontend/JS code to create proper subscription URLs pointing towards a JSON search result from ElasticSearch
- Subscription Titles (need to be autogenerated based on search results)

Views and URL-Schemes

The following are the URLs for the new subscription service:

URL	Description
-----	-------------

<code>/verify/<email>/<key></code>	verify the given email adress for the subscription with the given hash-key
<code>/subscribe</code>	subscribe an email adress to a given URL (email and url must be in POST-data)
<code>/unsubscribe/<email>/<key></code>	unsubscribe email-address from subscription with the given hash-key
<code>/list/<email>/<key></code>	list all subscriptions for the given email with the given verification key (for the user/email itself, not for any subscriptions)
<code>/list/<email></code>	re-send subscription list email, with the hash-key for the list view

Email Templates

The emails being sent are saved as `*.email` templates in `offenesparlament/templates/subscription/emails`; they follow the normal Django templating language. Variables can be defined and accessed via normal `{{ var_name }}` statements.

To facilitate the easy use of templates, an Email sending controller has been implemented in `offenesparlament.constants.EmailController`. It serves as a base class for the actual Email-Template-Classes (cf. for example `offenesparlament.constants.EMAIL.VERIFY_SUBSCRIPTION`). It provides an easy shortcut to Django's Email-Sending Module, automatically rendering the assigned template files and sending the email to the requested recipient.

API

This section describes implementation details for the API.

General & Structure

The current implementation provides api access to the scraped data at <http://offenesparlament.at/api>.

While following general REST guidelines, this is a purely read-only, no-auth API and provides only minimal documentation and support.

Implementation

The implementation is done via the [Django Rest Framework](#). The relevant implementation can be found at `offenesparlament/vies/api.py`.

Three custom base classes have been defined to extend the Django Rest Frameworks `ReadOnlyModelViewSet` and `HyperlinkedModelSerializer` respectively. The `ESViewSet` Class replaces the standard database queriesets with `ElasticSearch` queries to retrieve list data for the two main models 'Person' and 'Law' via ES for performance reasons. The `PaginatedFilteredViewSet` class provides a way of returning paginated results and also allowing to filter the result fields based on different serializers for list and detail views. And finally, the `DynamicFieldsModelSerializer` class extends the `HyperlinkedModelSerializer` to provide hyperlinked urls in the results and also allow per-request filtering of result fields to limit large models in the list view.

All ViewSets and Serializers for the Django models are defined subsequently in the same file. They all take the following form:

```
### MODEL: Category ###

class CategorySerializer(DynamicFieldsModelSerializer):
    """
    The category of a law (and all its sub-forms like pre-laws or inquiries) or
    of an opinion.
    """

    api_url = serializers.HyperlinkedIdentityField(
        view_name="op_api:Category-detail")

class Meta:
```

```

    model = Category
    fields = ('pk', 'title', 'api_url')

class CategoryViewSet(PaginatedFilteredViewSet):
    """
    The category of a law (and all its sub-forms like pre-laws or inquiries) or
    of an opinion.
    """
    model = Category
    queryset = Category.objects.all()
    serializer_class = CategorySerializer

```

Where necessary, a separate ListSerializer with different fieldsets or behaviour can be defined:

```

### MODEL: Debate ###

class DebateSerializer(DynamicFieldsModelSerializer):
    """
    A parliamentary debate (either in the Nationalrat or the Bundesrat).

    It is comprised of debate statements (shown only in the debate detail view
    for brevity and performance reasons).
    """

    llp = LegislativePeriodSerializer(
        required=True, fields=('pk', 'roman_numeral', 'number'))
    debate_statements = DebateStatementSerializer(required=False, many=True)

    api_url = serializers.HyperlinkedIdentityField(
        view_name="op_api:Debate-detail")

    class Meta:
        model = Debate
        fields = (
            'pk',
            'date',
            'title',
            'debate_type',
            'protocol_url',
            'detail_url',
            'nr',
            'llp',
            'debate_statements',
            'api_url'
        )

class DebateListSerializer(DynamicFieldsModelSerializer):
    """
    A parliamentary debate (either in the Nationalrat or the Bundesrat).

    It is comprised of debate statements (shown only in the debate detail view
    for brevity and performance reasons).
    """

    llp = LegislativePeriodSerializer(
        required=True, fields=('pk', 'roman_numeral', 'number'))

    api_url = serializers.HyperlinkedIdentityField(
        view_name="op_api:Debate-detail")

```

```
class Meta:
    model = Debate
    fields = (
        'pk',
        'api_url',
        'date',
        'title',
        'debate_type',
        'protocol_url',
        'detail_url',
        'nr',
        'llp',
    )

class DebateViewSet(PaginatedFilteredViewSet):
    """
    A parliamentary debate (either in the Nationalrat or the Bundesrat).

    It is comprised of debate statements (shown only in the debate detail view
    for brevity and performance reasons).
    """
    model = Debate
    queryset = Debate.objects.all()
    serializer_class = DebateSerializer
    list_serializer_class = DebateListSerializer
```

Finally, the internal routing mechanisms of the Django REST framework are defined at the bottom of the `offenesparlament.views.api` module by registering each viewset:

```
router = routers.DefaultRouter()
router.register(r'persons', PersonViewSet, base_name="Person")
router.register(r'laws', LawViewSet, base_name="Law")
router.register(r'debates', DebateViewSet, base_name="Debate")
[...]
```

Caveats

Performance: Currently, no performance improving measures beyond the use of ElasticSearch, where applicable, have been taken. Should the API prove to be too slow or overwhelm the server, request throttling should be employed. The Django REST Framework provides options for this.

Related `api_urls` for ElasticSearch-based viewsets must currently be index beforehand; deployment of the API thus requires a complete rebuilding of the index.

Existing Scrapers

Inquiries

General notes about the structure of the inquiries

- Inquiries can be submitted by members of the national assembly or federal assembly
- Inquiries can be directed at members of the Austrian government, the president of the court of auditors, the president of the national assembly or the chairmen of parliamentary committees
- Generally, inquiries are submitted in written form, with the exception of oral inquiries to the Austrian government

- Urgent inquiries (dringliche Anfragen) are oral inquiries with a written statement that is presented before the corresponding respondent and are structured in a special way to account for the discussion that is planned after the stakeholders have made their cases.
- While the structure of the data in the database is completely flat, there is some structure inherent to the parliamentary inquiries that can be exploited in post-processing or front-end:

- **Each inquiry has only one response, but each response can have multiple inquiries that it responds to, for example:**

- https://www.parlament.gv.at/PAKT/VHG/XX/AB/AB_05184/index.shtml

- **Some inquiries are similar are equal to others, e.g. mass inquiries to different governmental departments.**

- https://www.parlament.gv.at/PAKT/VHG/XXV/J/J_06447/index.shtml

- https://www.parlament.gv.at/PAKT/VHG/XXV/J/J_06448/index.shtml

- https://www.parlament.gv.at/PAKT/VHG/XXV/J/J_06449/index.shtml

- There are multiple types of inquiries, each type has a name and a shorthand. Shorthands are used for referencing and identifying certain inquiries in general communication and URLs.

Type	Shorthand
Schriftliche Anfragen an die Bundesregierung	J
Schriftliche Anfragen an die Bundesregierung (Bundesrat	J-BR
Mündliche Anfragen an die Bundesregierung	M
Mündliche Anfragen (Bundesrat)	M-BR
Dokumentenanfragen betr. EU an die Bundesregierung	JEU
Schriftliche Anfragen an Ausschussvorsitzende	JPR
Schriftliche Anfragen an PräsidentInnen des Nationalrats	JPR
Schriftliche Anfragen an RechnungshofpräsidentInnen	J

Scraper Structure

The scraper starts out using the RSS feed of the inquiry overview site

- Site: <http://www.parlament.gv.at/PAKT/JMAB/index.shtml>
- Feed: <http://www.parlament.gv.at/PAKT/JMAB/filter.psp>

The function `get_urls` in `inquiries.py` iterates over NR/BR and all legislative periods (LLP) to collect the links that will be parsed. The code includes a debugging variable containing assorted links for testing changes on all types of inquiries. `get_urls` will take approx. 2-3 minutes to record all available links to inquiries, and outputs the number of inquiries to be scraped in the terminal. Full scraping and parsing of all inquiries will take about 1-3 hours, depending on your internet connection and database speed.

In the parser, information is extracted and written into an Inquiry object and foreign keys are attached. If the inquiry is urgent (dringliche Anfrage), a more elaborate steps-parser is called (it is *almost* the same as the one for laws) than when it is not urgent. When a link to a written response can be found in the inquiry's history, that link is recorded and handed over to the callback function when the parser terminates. At the end of the function, the Inquiry object is saved and the output indicates which inquiry was created or updated. The functions `parse_keywords`, `parse_docs`, `parse_steps` and `parse_parliamentary` steps are self-explanatory helper-functions, they are parsers pulled out of the main parser function for clarity.

The callback function for inquiry requests is very similar to the main parser function. 90% of it is merely a simplification of what the parser function does, because lots of restrictions/edge cases don't apply to the responses of inquiries, e.g. urgent inquiries where the response is always oral. At the end of the inquiry response parser, the created or updated response object is attached to the original inquiry and the inquiry item is again saved to the database.

Model structure

The Inquiry and InquiryResponse models are subclasses of the Law model, inheriting all of its properties and augmented by a few that are unique to inquiries, specifically senders/receivers and responses.

Example Inquiry

- **title**
 - “Gender Budgeting” im BMEIA
- **source_link**
 - https://www.parlament.gv.at/PAKT/VHG/XXV/JJ_06440/index.shtml
- **parl_id**
 - J_06440
- **legislative_period**
 - XXV
- **description**
 - Schriftliche Anfrage der Abgeordneten Mag. Aygül Berivan Aslan, Kolleginnen und Kollegen an den Bundesminister für Europa, Integration und Äußeres betreffend “Gender Budgeting” (im BMEIA (Bundesministerium für Europa, Integration und Äußeres))
- **category**
 - Schriftliche Anfrage
- **keywords**
 - Bundeshaushalt III. Sonstiges
- **documents**
 - https://www.parlament.gv.at/PAKT/VHG/XXV/JJ_06440/imfname_465628.pdf
 - https://www.parlament.gv.at/PAKT/VHG/XXV/JJ_06440/fname_465724.pdf
 - https://www.parlament.gv.at/PAKT/VHG/XXV/JJ_06440/fnameorig_465724.html
- **sender**
 - Mag. Aygül Berivan Aslan
- **receiver**
 - Sebastian Kurz
- **steps**
 - 10.09.2015 Einlangen im Nationalrat (Frist: 10.11.2015)
 - 10.09.2015 Übermittlung an das Bundesministerium für Europa, Integration und Äußeres
 - 23.09.2015 91. Sitzung des Nationalrates: Mitteilung des Einlangens
 - 10.11.2015 Schriftliche Beantwortung (6251/AB)
- **response**
 - https://www.parlament.gv.at/PAKT/VHG/XXV/AB/AB_06251/index.shtml

Design

Design

Fonts

The font used throughout the design is called Source Sans Pro. It's the first open source font designed by Adobe. It was published under the SIL Open Font License. The source files can be downloaded via Sourceforge and Github. (Source: [Wikipedia](#))

The font can also be embedded with Google Fonts.

At the beginning, the Roboto font by Google was considered as an alternative. Both fonts – Source Sans Pro and Roboto – have very complete families, from very light to bold font styles.

We went for the Source Sans Pro because of her more distinctive character while being extremely readable. This font choice gives the website a modern look.

We use the font weights Regular and Bold. For emphasizing text, the italics provided with the font can be used. Please take care of using the real italics, not only slanting the regular font with ``.

Logo

The design of the logo is based on our font choice, Source Sans Pro. It is a combination of the sans serif fontface with the recently released Source Serif.

While the clean and simple sans serif shows our commitment for transparency, openness, modernity, the serif font stands for tradition. Both aspects mixed together are what makes this project special.

The icon reminds both the austrian flag (red-white-red) and a sending/transmitting symbol. The ending .at also represents austria and completes the url depicted in the logo: [offenesparlament.at](#)

Logo Download

The logo is available in the [most current formats for web and print applications](#):

- logo-offenesparlament_cmyk.pdf
- logo-offenesparlament_rgb.pdf
- logo-offenesparlament_rgb.png

Negative versions of the logo are also provided:

- logo-offenesparlament_cmyk_neg.pdf
- logo-offenesparlament_rgb_neg.pdf
- logo-offenesparlament_rgb_neg.pdf

The icon can be used separately in some cases:

- logo-icon-offenesparlament_rgb.png

Colors

The design comes with a colorscheme that is both colorful and soft. We often have long pages with a lot of text and tables, so the colors needed to work for structuring the site. It should look friendly, not too technical – we want to attract users with all kind of backgrounds, and therefore need the site to look human/welcoming. And last but not least, the colors needed to display the variety of the austrian parties.

The main color (used in the navigation bar throughout the site) is blue. It is also used for the background of the homepage in a slightly lighter hue.

An overview on the colors used on the site (in RGB, the percentages refer to the original color):

- Text color black 0/0/0
- Linkcolor blue 0/111/213
- Linkcolor blue hover in Header and Footer (50%) 127/183/234
- Blue Header Background 30/69/99

Design

- Grey Footer (85%) 38/38/38
- Light Red 245/95/87
- Red 236/55/51
- Red 50% for icons in the search bar 245/155/153
- Grey 50% for icons in the search bar, grey text, etc. 127/127/127
- Grey text footer (25%) 191/191/191
- Blue background homepage (95%) 41/78/107
- Grey search bar (10%) 229/229/229
- Grey text entry field in search bar (18%) 209/209/209
- Blue background lightbox 30/69/99 with opacity 0.9
- Grey lines in tables (15%) 217/217/217

Parties: - ÖVP 65/60/77 - Grüne = green 63/146/91 - SPÖ = red 236/55/51 - FPÖ 94/148/186 - Neos = pink 215/76/107 - Stronach = yellow 243/188/64 - BZÖ = orange 243/120/62

Other: - Green box element (Top10, etc.)

- fill color (10%) 235/244/238
- border color (25%) 207/228/214
- Light red box element (subscription form) - fill color (5%) 254/247/246 - border color (20%) 253/223/221
- Blue box element (laws) - border color and fill color lower part (15%) 221/227/232 - fill color upper part (5%) 244/246/247
- Buttons - border color light red 245/95/87 and fill color 255/255/255 (text and icon in light red) - hover: border and fill color 245/95/87 (text and icon changes to white)

Grid

Text

We have a number of basic text formatting covered, such as headlines, lists, etc ...

Tables

Tables always start with a white header row and bold text. All the other rows are alternating a white and light grey background (5% of blue 30/69/99). The rows are separated by light grey horizontal lines. There are no vertical lines or other elements to reinforce the columns. This way, the tables look simple, yet organised.

Text in tables is smaller than normal paragraph text. Icons can be used as needed. There is a generous padding in the table fields.

Links

Links within the content are formatted in blue and change opacity when hovered.

Icons

offenesparlament.at is a very text and table-heavy site. While more visualisations are part of our wishlist and might be realized in a next step, we needed smaller visual elements to help our readers and make the content easier to understand. That is where colors and, just as important, icons come in.

We used symbols that are based on the Streamline Icon Set (Line Version). To fit our needs, some icons were edited or created from scratch and added, such as the parliament icon.

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Community

If you'd like to stay in touch, sign up to our mailing list:

https://lists.metalab.at/mailman/listinfo/offenesparlament_at