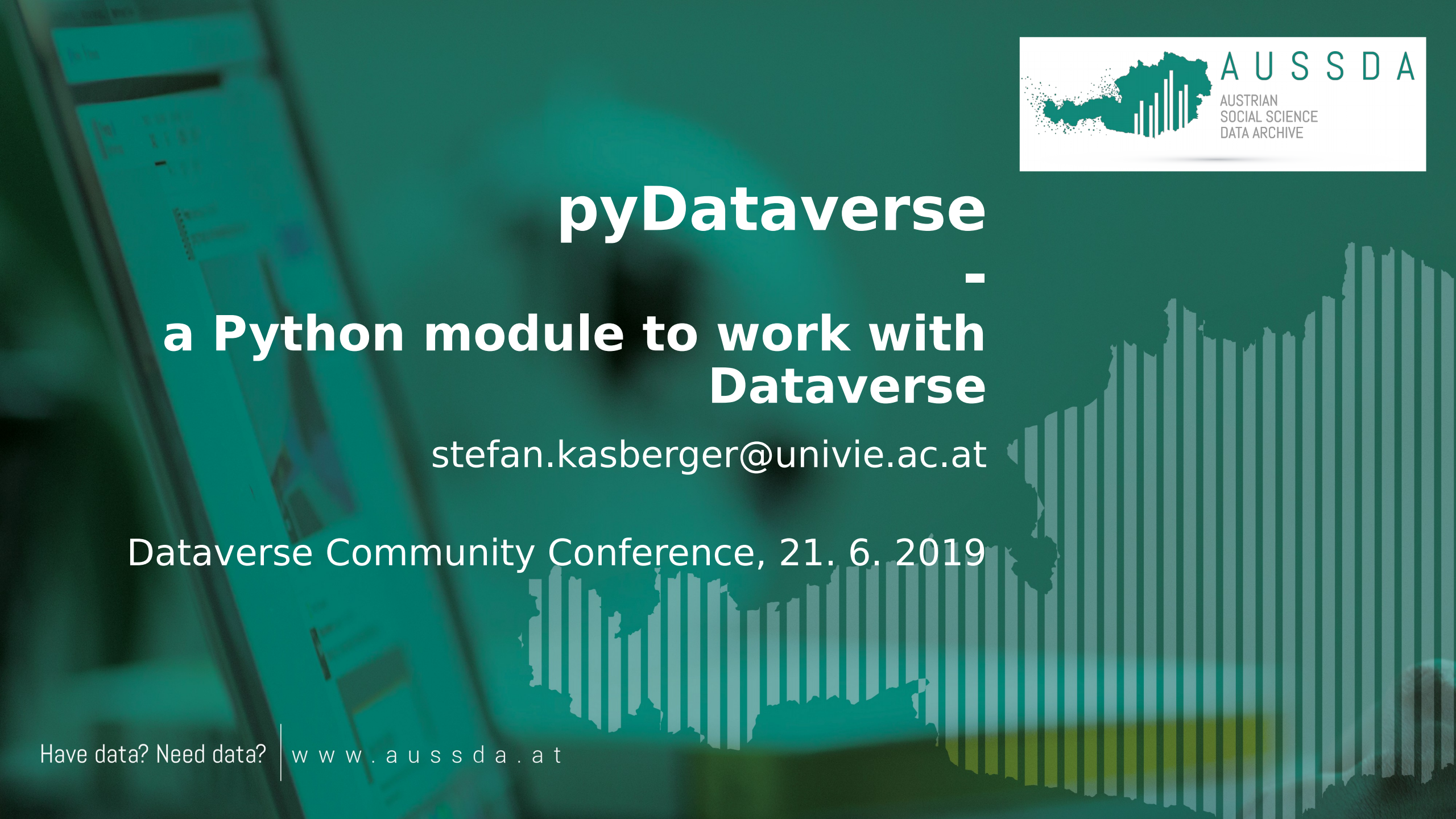


pyDataverse
-
a Python module to work with Dataverse

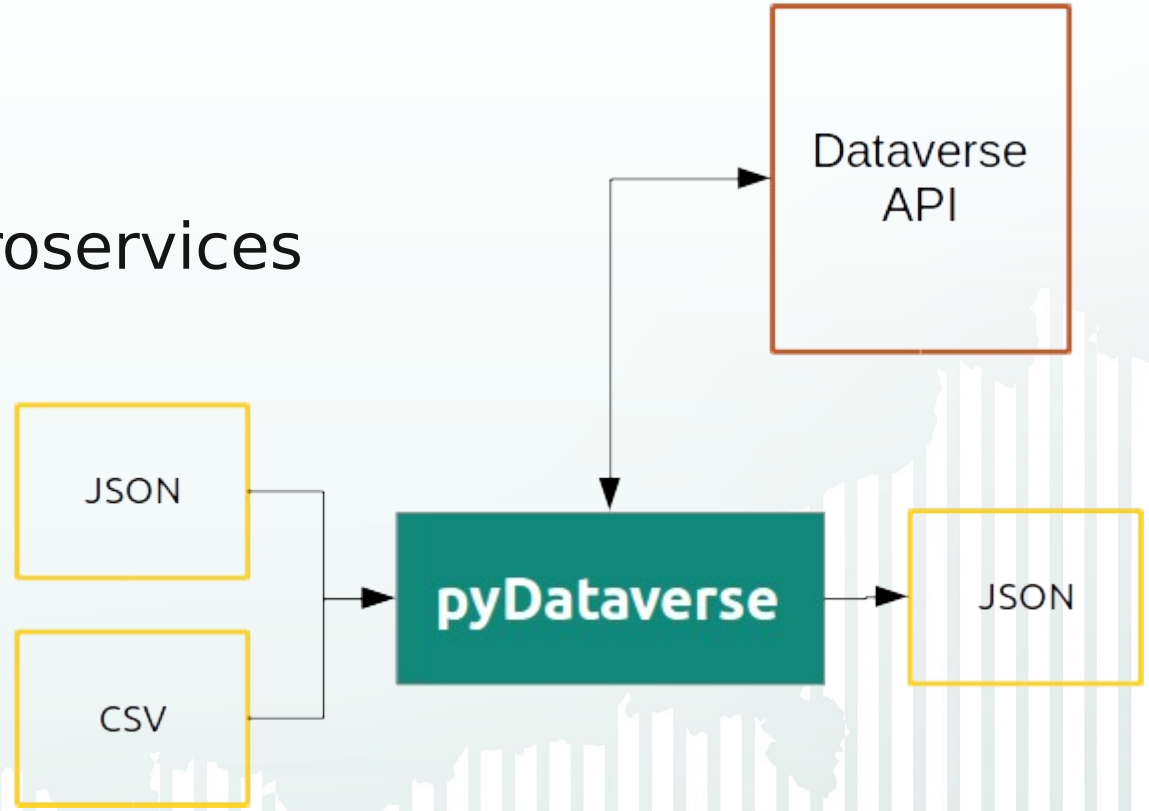
stefan.kasberger@univie.ac.at

Dataverse Community Conference, 21. 6. 2019

“pyDataverse is an open source Python module for Dataverse. It helps with Dataverse's data and accesses it's Api.”

Motivation

- Data migrations
- Support all users → Microservices
- Audience: Devs



Status Quo

- Version = 0.2.1
- PyPI: pip install pyDataverse
- Python 2 & 3
- Tests: pytest, tox, coveralls.io and travis ci
- Docs: Sphinx and ReadTheDocs
- Open Source: GitHub & MIT

Core-Features:

- Api wrapper
- Data models
- Utils
- Exceptions

Planned Activities

- Add Api requests
- Add data converters: DDI XML, GESIS DSpace, ???
- Microservice: NESSTAR data migration
- DVTree

Naming Conventions:

- Dataverse: dv_IDENTIFIER, prefix dv_, id = alias
- Dataset: ds_IDENTIFIER, prefix ds_, id = id
- Datafile: FILENAME

```
├── dv_harvard/
│   ├── metadata.json
│   └── dv_iqss/
│       ├── metadata.json
│       └── ds_microcensus-2018/
│           ├── metadata.json
│           └── datafiles/
│               ├── documentation.pdf
│               └── data.csv
```

Planned Activities

- Add Api requests
- Add data converters: DDI XML, GESIS DSpace, ???
- Microservice: NESSTAR data migration
- DVTree
- **Get people involved!**

AUSSDA
AUSTRIAN
SOCIAL SCIENCE
DATA ARCHIVE

Demo

```
{
  "datasetVersion": {
    "metadataBlocks": {
      "citation": {
        "fields": [
          {
            "value": "Darwin's Finches",
            "typeClass": "primitive",
            "multiple": false,
            "typeName": "title"
          },
          {
            "value": [
              {
                "authorName": {
                  "value": "Finch, Fiona",
                  "typeClass": "primitive",
                  "multiple": false,
```

```
In [10]: from pyDataverse.models import Dataset
    ...: ds = Dataset()
    ...: ds.import_metadata('tests/data/dataset_min.json', 'dv_up')
    ...: ds.title
    ...:
geospatial not in json
socialscience not in json
journal not in json
Out[10]: "Darwin's Finches"

In [11]:
```

```
In [11]: ds.dict()
Out[11]:
{'datasetVersion': {'metadataBlocks': {'citation': {'displayName': 'Ci
tation Metadata',
    'fields': [{'typeName': 'title', 'value': "Darwin's Finches"},
     {'typeName': 'subject', 'value': ['Medicine, Health and Life Scie
nces']},
     {'typeName': 'kindOfData', 'value': []},
     {'typeName': 'relatedMaterial', 'value': []},
     {'typeName': 'relatedDatasets', 'value': []},
     {'typeName': 'otherReferences', 'value': []},
     {'typeName': 'dataSources', 'value': []},
     {'typeName': 'kindOfData', 'value': []},
     {'typeName': 'timePeriodCovered', 'value': []},
     {'typeName': 'dateOfCollection', 'value': []},
     {'typeName': 'software', 'value': []},
     {'typeName': 'datasetContact',
      'value': [{'datasetContactEmail': {'typeName': 'datasetContactEm
ail',
         'value': 'finch@mailinator.com'},
        'datasetContactName': {'typeName': 'datasetContactName',
         'value': 'Finch, Fiona'}}]},
     {'typeName': 'keyword', 'value': []},
```

```
In [14]: from pyDataverse.api import Api
    ...: api = Api('http://localhost:8085', '98f04923-f77d-43e6-9445-d
    ...: f8636687f89')
    ...: api.status
    ...:
Out[14]: 'OK'

In [15]:
```

```
In [17]: resp = api.get_dataverse('root')
    ...: resp.json()
    ...:
Out[17]:
{'data': {'alias': 'root',
  'creationDate': '2019-04-11T10:37:54Z',
  'creator': {'affiliation': 'Dataverse.org',
   'authenticationProviderId': 'builtin',
   'createdTime': '2019-04-11T10:37:53Z',
   'displayName': 'Dataverse Admin',
   'email': 'dataverseAdmin@mailinator.com',
   'firstName': 'Dataverse',
   'id': 1,
   'identifier': '@dataverseAdmin',
   'lastApiUseTime': '2019-04-11T10:37:54Z',
   'lastLoginTime': '2019-04-11T10:37:53Z',
   'lastName': 'Admin',
   'persistentUserId': 'dataverseAdmin',
   'position': 'Admin',
   'superuser': False},
  'dataverseContacts': [],
  'dataverseType': 'UNCATEGORIZED',
  'description': 'The root dataverse.',
```

```
In [21]: resp = api.create_dataset('root', ds.json())

In [22]:
```

AUSSDA
AUSTRIAN
SOCIAL SCIENCE
DATA ARCHIVE

Get Involved!

Have data? Need data? | www.aussda.at

Get involved

- Use pyDataverse
- Give feedback
- Contribute to the development
- Share it

Get involved

- Use pyDataverse
- Give feedback
- Contribute to the development
- Share it

No contribution is too small!

How to contribute

- Go to pyDataverse's documentation
  - https://pydataverse.readthedocs.io
- Improve Documentation
- Issues: Feature requests, bug reports, pull requests
- New requests
- **DDI XML import/export !**
- Bigger additions: Please get in touch, before you start.

Contact

- Email: stefan.kasberger@univie.ac.at
- GitHub: aussda/pyDataverse
- Website: aussda.at/en
- Twitter: @theAUSSDA

I would like to acknowledge and thank the following people for their support and feedback: Lars Kaczmirek, Vyacheslav Tykhonov, Philip Durbin and Danny Brooke.

Thank you!

info@aussda.at
+43 01 4277 15323
www.aussda.at