
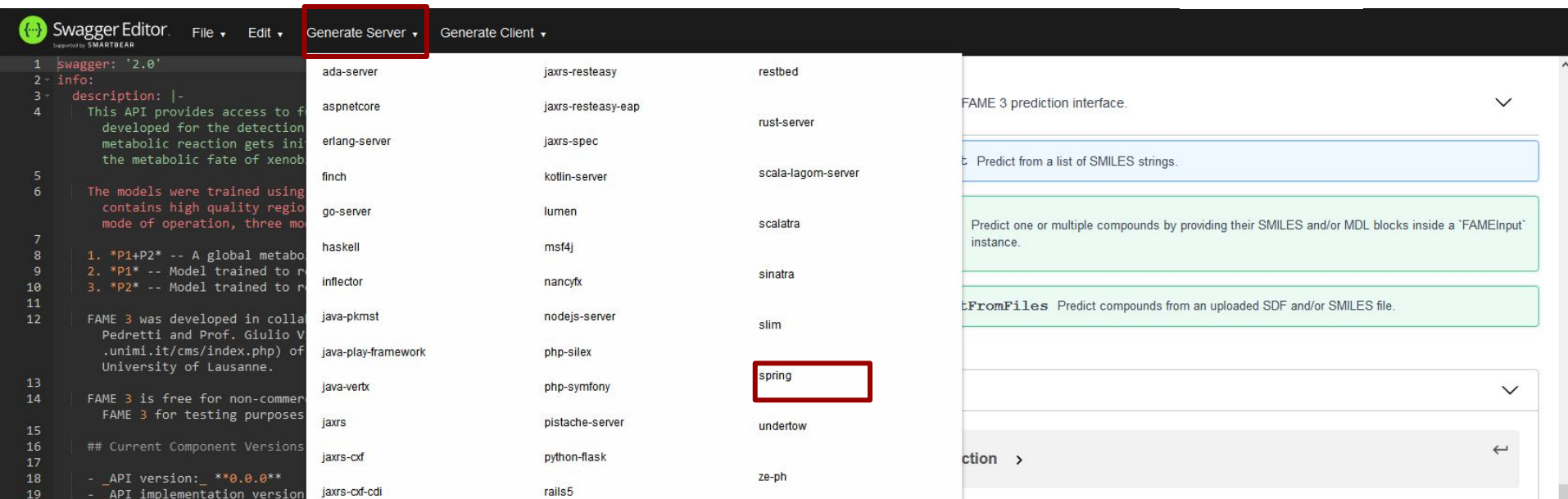


FAME 3 API in OpenRiskNet

Martin Šícho

> FAME 3 API - implementation

- API designed according to the OpenAPI 2.0 standard (fka Swagger RESTful API Documentation Specification)
- Original FAME 3 software package in Java => 



The screenshot displays the Swagger Editor interface. The 'Generate Server' dropdown menu is open, showing a list of server implementations. The 'spring' option is highlighted with a red box. The background shows the Swagger Editor's code editor with a Swagger 2.0 specification for the FAME 3 API, including a description and a list of endpoints.

```
1 swagger: '2.0'
2 info:
3   description: |-
4     This API provides access to f
5     developed for the detection
6     metabolic reaction gets ini
7     the metabolic fate of xenob
8
9     The models were trained using
10    contains high quality regio
11    mode of operation, three mo
12
13    1. *P1+P2* -- A global metabo
14    2. *P1* -- Model trained to r
15    3. *P2* -- Model trained to r
16
17    FAME 3 was developed in colla
18    Pedretti and Prof. Giulio V
19    .uniml.it/cms/index.php) of
20    University of Lausanne.
21
22    FAME 3 is free for non-commen
23    FAME 3 for testing purposes
24
25    ## Current Component Versions
26
27    - _API version:_ **0.0.0**
28    - _API implementation version
```

| | | |
|---------------------|--------------------|--------------------|
| ada-server | jaxrs-resteasy | restbed |
| aspnetcore | jaxrs-resteasy-eap | rust-server |
| erlang-server | jaxrs-spec | scala-lagom-server |
| finch | kotlin-server | scalatra |
| go-server | lumen | sinatra |
| haskell | msf4j | slim |
| inflector | nancyfx | spring |
| java-pkms | nodejs-server | undertow |
| java-play-framework | php-silex | ze-ph |
| java-vertx | php-symfony | |
| jaxrs | pistache-server | |
| jaxrs-cxf | python-flask | |
| jaxrs-cxf-cdi | rails5 | |

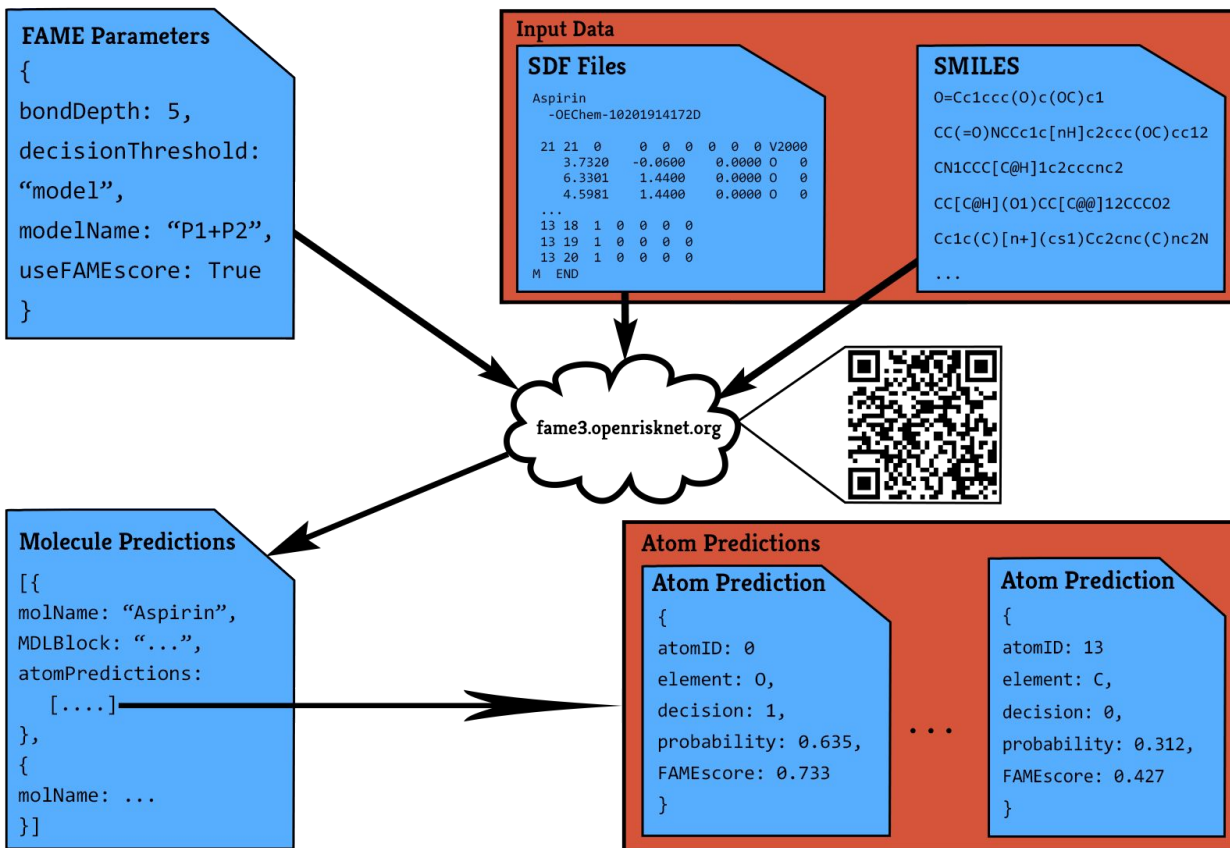
FAME 3 prediction interface.

Predict from a list of SMILES strings.

Predict one or multiple compounds by providing their SMILES and/or MDL blocks inside a 'FAMEInput' instance.

FromFiles Predict compounds from an uploaded SDF and/or SMILES file.

> FAME 3 API - using it



```
import requests
```

```
parameters = {  
    'modelName' : 'P1P2'  
    , 'bondDepth' : "DEPTH_5"  
    , 'useFAMEScore' : True # turn this off for faster calculation  
    , 'smiles' : cpds["SMILES"].tolist()  
    , 'decisionThreshold' : "model"  
}
```

```
resp = requests.get(  
    url + '/prediction'  
    , params=parameters  
    , headers={"accept" : "application/json"}  
)
```

```
if resp.ok:  
    json_data = resp.json()  
    for prediction in json_data['predictions']:  
        process_molecule(prediction)  
else:  
    print("Prediction failed...")  
    process_error(resp)
```