

Parliament Sentiment Radar

DownloadMetaData

DownloadZip

```
+ download(String link,
String dir);
```

ScrapePictures

```
- String searchUrl;
- String[] pictureAndData;
```

```
+ ScrapePictures(String
name, String surname)
+ getSearchUrl(String name,
String surname);
+ performSearch();
+ getMetaData();
```

UnzipFile

```
+ unzip(String source,
String target);
+ zipSlipProject(ZipEntry
zipEntry, Path targetDir);
```

DownloadXml

```
+ load(int nameOfFile, int
period, String
linkToLoad);
+ loadDTD();
```

Parse

```
+ pars();
```

ScrapePictures

```
- String link;
- File out;
```

```
+ protocolLoading (String link,
File out);
+ run();
```

database

MongoDB-ConnectionHandler

```
- MongoClient mongoClient;
- MongoDB database;
- MongoCollection <Document> collection

+ close();
+ deleteAllCollections();
+ getDatabase();
+ getMongoClient();
+ getCollection();
+ getCollection(String collectionName);
+ getDBDocument(String id, String
collectionName)
+ uploadAllProtocols (ArrayList<Protocol>
protocolList)
+ insertMember(Member member);
+ insertSpeech(Speech speech);
+ insertComment(Comment comment)
+ updateSpeech(Speech speech);
+ updateComment(Comment comment);
+ deleteDocumentOnDB(String id, String
collectionName) ;
+ countQuery(String fieldName, int value,
String collectionName)
+ countQuery(String fieldName, String
value, String collectionName);
+ joinQuery(String collectionName, String
from, String localField, String foreignField,
String as, String fieldName, String value);
+ joinQuery(String collectionName, String
from, String localField, String foreignField,
String as, String fieldName, int value);
+ joinQuery(String collectionName, String
from, String localField, String foreignField,
String as);
+ aggregate(String collectionName);
```

BTOBJECTTo-MongoDocument

```
+ Document createMongo-
Document (Member member)
+ Document createMongo-
Document (Speech speech)
+ Document createMongo-
Document (Comment comment)
+ Sentiment getDocument-
Sentiment(JCas jCas)
```

PropertiesFile

```
+ readPropertiesFile();
```

ApiConnection

```
- MongoDBConnectionHandler
handler
- List<Strings> allParties

+ main(Strings[] args)
+ processQuery()
+ getSpeechAttributeSpeaker()
+ getSpeechAttributeParty()
+ sortMap()
```

nlp

NLP

```
- AnalysisEngine pAE = null;
- AggregateBuilder pipeline;

+ getPipeline();
+ analyse(String text);
```

data

<<interface>> Protocol

```
+ void initialize(Document doc);
+ void setStartPageNr(Integer startPageNr);
+ int getStartPageNr();
+ void setDate(String pDate);
+ String getDate();
+ void setAgendaItem(Document doc);
+ ArrayList<AgendaItem>
getAllAgendaItems();
+ void setSessionID(Integer sessionID);
+ Integer getSessionID();
+ void setLeaders (Document doc);
+ ArrayList<String> getLeaders();
+ void setElectionPeriod (Document doc);
+ Integer getElectionPeriod();
+ void setTitle(String title);
+ String getTitle();
```

<<interface>> AgendaItem

```
+ void setAgendaItemID(String id);
+ String getAgendaItemID();
+ void setAllSpeeches(Node nNode, Protocol
protocol);
+ ArrayList<Speech> getSpeeches();
```

<<interface>> Member

```
+ String setName(Node Mdb);
+ String setSurname(Node Mdb);
+ String setParty(Node Mdb);
+ Date setBirthDate(Node Mdb);
+ String setID(Node Mdb);
+ String setPlaceOfBirth(Node Mdb);
+ String setCountryOfBirth(Node Mdb);
+ Date setDateOfDeath(Node Mdb);
+ String setGender(Node Mdb);
+ String setMaritalStatus(Node Mdb);
+ String setReligion(Node Mdb);
+ String setOccupation(Node Mdb);
+ String[] setMetaData(String name,
String surname);
+ String getName();
+ String getSurname();
+ Date getBirthDate();
+ String getID();
+ String getPlaceOfBirth();
+ String getCountryOfBirth();
+ Date getDateOfDeath();
+ String getGender();
+ String getMaritalStatus();
+ String getOccupation();
+ String getReligion();
+ String[] getMetaData();
+ String getFullInfoForTesting();
```

<<interface>> Speech

```
+ void setSpeechID(String speechID);
+ void initialize(Node kNode, Protocol protocol,
AgendaItem agendaItem);
+ String getSpeechID();
+ String getSpeakerID();
+ void setSpeakerID(String speakerID);
+ String getText();
+ ArrayList<Comment> getAllComments();
+ Protocol getProtocol();
+ void setProtocol(Protocol protocol);
+ AgendaItem getAgendaItem();
+ void setAgendaItem(AgendaItem agendaItem);
```

<<interface>> Comment

```
+ String getCommentID();
+ void setCommentID(String id);
+ String getText();
+ void setText(String text);
+ String getSpeech();
+ void setSpeech(String id);
```

helper

InitializeProtocols

```
- ArrayList<Protocol> allProtocols
- XMLFileReader xml;
- ArrayList<Document> allXmlArrayList

InitializeProtocols()
```

XMLFileReader

```
- ArrayList<Document> allXmlArrayList

+ getAllFiles();
+ getMetadataXml();
+ ProcessFile(File file);
```

impl

ProtocolFileImpl

```
- String pDate;
- Integer electionPeriod;
- Integer sessionID;
- String title;
- Integer startPageNr;
- ArrayList<AgendaItem> allAgendaItems;
- ArrayList<String> leaders;

+ initialize(Document doc)
```

AgendaItemFile Impl

```
- String agendaItemID;
- ArrayList<Speech> allSpeeches;
```

SpeechFile Impl

```
- String speechID;
- String text = "";
- ArrayList<Comment> allComments;
- String speakerID;
- private Protocol protocol = null;
- private AgendaItem agendaItem = null

+ initialize(Node kNode, Protocol protocol,
AgendaItem agendaItem)
```

MemberFile Impl

```
- String name;
- String surname;
- String party;
- Date birthDate;
- String[] metaData;
- String id;
- String placeOfBirth;
- String countryOfBirth;
- Date dateOfDeath;
- String gender;
- String maritalStatus;
- String religion;
- String occupation;

+ MemberFile_Impl(Node Mdb)
```

CommentFile Impl

```
- private String id;
- private String text;
- private String speechID;

+ hashCode();
```