org.ppr.abschlussprojekt

data

interface

«interface»
Protokoll

+getProtocolId():String
+getProtocolIndex():int
+getProtocolTitle():String
+getDate():Date
+getProtocolStart():Timestamp
+getProtocolEnd():Timestamp
+getWahlperiode():int
+getPlace():String
+getAgendaList():List
+addAgenda(agenda:Agenda)
+getSpeakerMap():Map
+setSpeakerMap()
+setSpeakerMap()

«interface»
Agenda

+getAgendaId():String
+setAgendaId(agendaId:String)
+getAgendaIndex()
+getSpeechList():List
+getProtocol():Protocol
+getSpeakerMap():Map
+addSpeech(speech:Speech)
+getCommentMap():Map
+addCommentMap(speech:Speech)
+addSpeaker()

<<interface>>
Speaker

+getSpeakerId():String
+getVorname():String
+getNachname():String
+getTitel():String
+getBirthDate():long
+getBirthplace():String
+getGender():String
+getReligion():String
+getProfession():String
+getAcademicTitle():String
+getFamilyStatus():String
+getDeathDate:long
+getFraction():Fraction
+setFraction(fraction:Fraction)
+getParty():Party
+setParty(party:Party)
+getSpeechMap():Map
+getCommentList():List
+getRoll():String
+getProtocolList():List
+getAgendaList():List
+addSpeech(speech:Speech)
+addAgenda(agenda:Agenda)
+addProtocol(protocol:Protocol)
+getFractionName():String
+getPartyName():String
+setPartyName(partyName:String)
+setFractionName(fractionName:String)
+getSpeechList():List
+getPictureInfoList():List

<<interface>>
Speech

+getSpeechId():String
+setSpeechId(speechId:String)
+getText():String
+setText(text:String)
+getSpeaker():Speaker
+setSpeaker(speaker:Speaker)
+getCommentList():List
+getProtocol():Protocol
+setProtocol(protocol:Protocol)
+getAgenda():Agenda
+setAgenda(agenda:Agenda)
+setAllCommentAttributes()
+addSpeakerProtocol()
+addSpeakerAgenda()
+getFraction():Fraction
+setFraction(fraction:Fraction)
+getParty():Party
+setParty(party:Party)
+setCommentFraction()
+setCommentParty()
+getFractionName():String
+getPartyName():String
+setPartyName(partyName:String)

<<interface>>
Comment

+getCommentSpeechId():String
+getContent():String
+setContent(content:String)
+getSpeech():Speech
+setSpeech(speech:Speech)
+getFraction():Fraction
+setFraction(fraction:Fraction)
+getSpeaker():Speaker
+setSpeaker(speaker:Speaker)
+getCommentIndex:int
+setCommentIndex(commentIndex:int)
+getParty():Party
+setParty(party:Party)

<<interface>>
Party

+getPartyName():String
+getSpeakerList():List
+setPartyName(partyName:String)
+setSpeakerList(speakerList:List)
+getProtocolList():List
+setProtocolList(protocolList:List)
+addSpeaker(speaker:Speaker)
+addProtocol(protocol:Protocol)
+getAllSpeechList():List
+getAllCommentList():List
+getNumComment():int

<<interface>>
Fraction

+getFractionName():String
+setFractionName(fractionName:String)
+setSpeakerList(speakerList:List)
+getSpeakerList():List
+getProtocolList():List
+setProtocolList(protocolList:List)
+addSpeaker(speaker:Speaker)
+addProtocol(protocol:Protocol)
+getAllSpeechList():List
+getAllCommentList():List
+getNumComment():int

<<interface>>
NLPtemplate

+setPersons(persons:List)
+addPerson(person:Note)
+getPersons():List
+setLocations(locations:List)
+addLocation(location:Node)
+getLocations():List
+setOrganisations(organisations:List)
+addOrganisation(organisation:Node)
+getOrganisations():List
+setTokens(tokens:List)
+getTokens():List
+setSentences(sentences:List)
+addSentence(sentence:Note)
+getSentences():List
+setSentiment(sentiment:double)
+getSentiment():double
+setLemmas(lemmas:List)
+addLemma(lemma:Note)
+getLemmas():list
+setDdc(ddc:List)
+addDdc(ddc:Note)
+getDdc():List
+setPOS(pos:Note)
+addPOS(pos:Note)
+getPOS():List
+setUima(uima:String)
+getUima():String

impl

file

Protocol_File_Impl

-protocolId:String
-protocolIndex:int
-protocolTitle:String
-date:Date
-protocolStart:Timestamp
-protocolEnd:Timestamp
-wahlperiode:int
-place:String
-agendaList:List
-speakerMap:Map

+Protocol_File_Impl(rootNode:Node)
-setUpAttributes(Node rootNode)
+toString():String
+getProtocolId():String
+getProtocolIndex():int
+getProtocolTitle():String
+getDate():Date
+getPlace():String
+getProtocolStart():Timestamp
+getProtocolEnd():Timestamp
+getWahlperiode():int
+getAgendaList():List
+addAgenda()
+getSpeakerMap():Map
+setSpeakerMap()

Agenda_File_Impl

-agendaId:String
-agendaIndex:String
-speechList:List
-protocol:Protocol
-commentMap:Map
-speakerMap:Map

+Agenda_File_Impl(Node agendaNode, Protocol protocol)
-setUpAttributes(Node agendaNode, Protocol protocol)
+toString():String
+getAgendaId():String
+setAgendaId(agendaId:String)
+getAgendaIndex():String
+getSpeechList():List
+getProtocol():Protocol
+getCommentMap():Map
+getSpeakerMap():Map
+addSpeech(Speech speech)
+addCommentMap(speech:Speech)
+addSpeaker(Speaker speaker, Map allSpeakerMap)

Speaker_File_Impl

-speakerId:String
-vorname:String
-nachname:String
-titel:String
-roll:String
-fractionName:String
-partyName:String
-fraction:Fraction
-party:Party
-birthDate:long
-deathDate:long
-birthplace:String
-gender:String
-religion:String
-profession:String
-academicTitle:String
-familyStatus:String
-pictureInfoList:List
-speechMap:Map
-commentList:List
-protocolList:List
-agendaList:List

+Speaker_File_Impl(Node speechNode, Map speakerStammdatenMap)
-setUpAttributes(Node speechNode, Map speakerStammdatenMap)
+toString():String
+getSpeakerId():String
+getVorname():String
+getNachname():String
+getTitel():String
+getRoll():String
+getBirthDate():String
+getBirthplace():String
+getGender():String
+getReligion():String
+getProfession():String
+getAcademicTitle():String
+getFamilyStatus():String
+getDeathDate():long
+getFraction():Fraction
+getParty():Party
+getCommentList():List
+getProtocol():Protocol
+getAgenda():Agenda
+addSpeakerProtocol()
+addSpeakerAgenda()
+setCommentFraction()
+setCommentParty()

Speech_File_Impl

-speechId:String
-protocol:Protocol
-agenda:Agenda
-text:String
-speechLength:int
-fraction:Fraction
-party:Party
-fractionName:String
-partyName:String
-speaker:Speaker
-commentList:List

+Speech_File_Impl(Node speechNode, Map speakerMap, Map commentMap, Map speakerStammdatenMap)
-setUpAttributes(Node speechNode)
+toString():String
+getFractionName():String
+getPartyName():String
+getSpeechId():String
+getText():String
+getSpeaker():Speaker
+getFraction():Fraction
+getParty():Party
+getCommentList():List
+getProtocol():Protocol
+getAgenda():Agenda
+setIndex(index:String)
+addSpeakerProtocol()
+addSpeakerAgenda()
+setCommentFraction()
+setCommentParty()

Comment_File_Impl

-commentSpeechId:String
-commentIndex:int
-content:String
-speech:Speech
-fraction:Fraction
-party:Party
-speaker:Speaker

+Comment_File_Impl(String content, int commentIndex, String speechId)
+toString():String
+setUpAttributesWithSpeech(speech:Speech)
+getCommentSpeechId():String
+getContent():String
+getSpeech():String
+getFraction():Fraction
+getSpeaker():Speaker
+getCommentIndex():int
+getParty():Party

Party_File_Impl

-partyName:String
-speakerList:List
-protocolList:List

+Party_File_Impl(String partyName)
+toString():String
+getPartyName():String
+getSpeakerList():List
+getProtocolList():List
+addSpeaker()
+addProtocol()
+getAllSpeechList():List
+getAllCommentList():List
+getNumComment():int

Fraction_File_Impl

-fractionName:String
-speakerList:List
-protocolList:List

+Fraction_File_Impl(String fractionName)
+toString():String
+getFractionName():String
+getSpeakerList():List
+getProtocolList():List
+addSpeaker(speaker:Speaker)
+addProtocol(protocol:Protocol)
+getAllSpeechList():List
+getAllCommentList():List
+getNumComment():int

SpeakerStammdaten

-speakerId:int
-vorname:String
-nachname:String
-titel:String
-roll:String
-fractionName:String
-partyName:String
-fraction:Fraction
-party:Party
-birthDate:long
-deathDate:long
-birthplace:String
-gender:String
-religion:String
-profession:String
-academicTitle:String
-familyStatus:String

+SpeakerStammdaten(Node mdbNode)
+getId():int
+getVorname():String
+getNachname():String
+getTitel():String
+getRoll():String
+getBirthDate():String
+getBirthplace():String
+getGender():String
+getReligion():String
+getProfession():String
+getAcademicTitle():String
+getFamilyStatus():String
+getDeathDate():long
+getFractionName():String
+getPartyName():String

mongodb

Speaker_MongoDB_Impl

-speaker:Speaker
-speakerId:String
-vorname:String
-nachname:String
-title:String
-fractionName:String
-partyName:String
-birthDate:long
-birthplace:String
-gender:String
-religion:String
-profession:String
-academicTitle:String
-familyStatus:String
-deathDate:long
-pictureInfoList:List
-speechIdList:List
-agendaIdList:List
-protocolIdList:List
-commentIdMap:Map
-document:Document

+Speaker_MongoDB_Impl(document:Document)
+Speaker_MongoDB_Impl(speaker:Speaker)
-setCommentarMap():Map
+getMongoDocument():Document
+getSpeakerId():String
+getVorname():String
+getNachname():String
+getTitel():String
+getFractionName():String
+getPartyName():String
+getBirthDate():long
+getBirthplace():String
+getReligion():String
+getProfession():String
+getAcademicTitle():String
+getFamilyStatus():String
+getDeathDate():long
+getPictureInfoList():List
+getSpeechIdList():List
+getAgendaIdList():List
+getProtocolIdList():List
+getCommentIdMap():Map
+getDownloadedDocument():Document
+toText():String

Speech_MongoDB_Impl

-speech:Speech
-speechId:String
-speakerId:String
-agendaDoc:Document
-protocolDoc:Document
-text:String
-commentIdList:List
-document:Document
-protocolDate:long
-protocolStartTime:long
-protocolEndTime:long
-protocolIndex:int
-protocolTitle:String
-protocolPlace:String
-protocolWp:int
-agendaIndex:String
-agendaId:String
-agendaTitle:String

+Speech_MongoDB_Impl(document:Document)
+Speech_MongoDB_Impl(speech:Speech)
+setProtocolDoc()
+setAgendaDoc()
+getMongoDocument():Document
+getSpeechId():String
+getSpeakerId():String
+getText():String
+getCommentIdList():List
+getDownloadedDocument():Document
+getProtocolDate():long
+getProtocolStartTime():long
+getProtocolEndTime():long
+getProtocolIndex():int
+getProtocolTitle():String
+getProtocolPlace():String
+getProtocolWp():int
+getAgendaIndex():String
+getAgendaId():String
+getAgendaTitle():String

Comment_MongoDB_Impl

-comment:Comment
-speechId:String
-text:String
-speechId:String
-speakerId:String
-document:Document

+Comment_MongoDB_Impl(document:Document)
+Comment_MongoDB_Impl(comment:Comment)
+getMongoDocument():Document
+getCommentId():String
+getText():String
+getSpeechId():String
+getSpeakerId():String
+getDownloadedDocument():Document

nlp

Agenda_NLP_Impl

-index:String
-id:String
-title:String

+Agenda_NLP_Impl(doc:Document)
+getAgendaId():String
+getTitle():String
+setTitle(title:String)
+setAgendaId(agendaId:String)
+getTitle():String
+getAgendaIndex():String
+setIndex(index:String)
+getSpeechList():List
+getProtocol():Protocol
+getSpeakerMap():Map
+addSpeech(speech:Speech)
+addCommentMap(speech:Speech)
+addSpeaker(speaker:Speaker)
+toText():String

Speech_NLP_Impl

-_id:String
-text:String
-speaker:String
-protocol:Document
-protocolDoc:Protocol_NLP_Impl
-comments:List
-agenda:Agenda_NLP_Impl
-persons:List
-locations:List
-organisations:List
-tokes:List
-sentences:List
-sentiment:double
-lemma:List
-ddc:List
-pos:List
-uima:String
-personIndices:List
-organisationIndices:List
-locationIndices:List

+getPersonIndices():List
+getOrganisationIndices():List
+getLocationIndices():List
+get_id():String
+getText():String
+getSpeaker():String
+getProtocol():Protocol_NLP_Impl
+getComments():List
+getAgenda():Agenda_NLP_Impl
+getPersons():List
+getLocations():List
+getOrganisations():List
+getTokens():List
+getSentences():List
+getTokens():List
+getSentiment():double
+getLemmas():List
+getDdc():List
+getPOS():List
+getUima()List
+toTex():String

Comment_NLP_Impl

-_id:String
-text:String
-speaker:String
-speech:String
-persons:List
-locations:List
-organisations:List
-token:List
-sentences:List
-sentiment:double
-lemma:List
-ddc:List
-pos:List
-uima:String

+get_id():String
+getText():String
+getPersons():List
+getLocations():List
+getOrganisations():List
+getTokens():List
+getSentences():List
+getTokens():List
+getSentiment():double
+getLemmas():List
+getDdc():List
+getPOS():List
+getUima()List
+toTex():String

Protocol_NLP_Impl

-date:long
-starttime:long
-endtime:long
-index:int
-wp:int
-title:String
-place:String

+getProtocolId():String
+getProtocolIndex():int
+getProtocolTitle():String
+getDate():Date
+getProtocolStart():Timestamp
+getProtocolEnd():Timestamp
+getWahlperiode():int
+getPlace():String
+getAgendaList():List
+getProtocol():Protocol
+getSpeakerMap():Map
+addAgenda(agenda:Agenda)
+getSpeakerMap():Map
+toTex(speeches:Speech):String

database

DataService

-dbHandler:MongoDBHandler

+DataService(dbHandler:MongoDBHandler)
+getTop(subject:String, resultLimit:int, startDate:long, endDate:long, collection:String, search:String):List
+getTopSpeaker(resultLimit:int, startDate:long, endDate:long, collection:String, search:String):List
+getSentiment(startDate:long, endDate:long, collection:String, search:String):List
+getCommentNetwork(limit:int, startDate:long, endDate:long):JSONObject
+commentNetByProtocol(protocol:String):JSONObject
+getCategoriesNetwork(limit:int, startDate:long, endDate:long):JSONObject
+categoriesByProtocol(protocol:String):JSONObject
+getCategoryBySpeech(limit:int, startDate:long, endDate:long):JSONObject
+categorySpeechProtocol(protocol:String):JSONObject
+getRelatedSpeeches(collection:String, search:String, converter:Converter):List
+getSpeakerData(db:MongoDBHandler, speakerID:String):Speaker_MongoDB_Impl

MongoDBHandler

-mongoClient:MongoClient
-mongoDatabase:MongoDatabase
-converter:Converter

+MongoDBHandler()
+getCollection(collection:String):MongoCollection
+getCollectionDocument(collection:String):MongoCollection
+insertOne(document:Document, collectionName:String)
+insertOneNotExists(document:Document, collectionName:String)
+readAndFilter(collection:MongoCollection, query:BasicDBObject):List
+count(collection:MongoCollection):long
+doesExist(collection:MongoCollection, _id:String):boolean
+doesExist(collection:MongoCollection, _id:String):boolean
+updateWithResult(collection:String, query:Document, updateDocument):String
+update(collectionName:String, query:Document, updateDocument:Document)
+delete(query:BasicDBObject, collection:MongoCollection)
+doQueryIterator(query:BasicDBObject, sCollection:String):MongoCursor
+doSortedQueryIterator(collection:String, query:Document:Document, sortObject:BasicDBObject):MongoCursor
+getMongoCursor(cursor:String):MongoCursor
+getMongoCursor(collection:MongoCollection):MongoCursor
+getOneDocument(collection:MongoCollection, id:String):Document
+aggregateDocument(optionList:List, collectionName:String):List
+updateCountProgress(docId:Name:String, keyName:String, num:int)
+resetCountProgress(progressName:String)

MongoDBHelper

+insertAllSpeakerDocument(speakerMap:Map, dbHandler:MongoDBHandler, collectionName:String)
+insertAllSpeechDocument(speechMap:Map, dbHandler:MongoDBHandler, collectionName:String)
+insertAllCommentDocument(commentMap:Map, dbHandler:MongoDBHandler, collectionName:String)
+insertAllDocument(dbHandler:MongoDBHandler)

helper

XMLHelper

XMLDownloader

+downloadAllXML():List
+downloadTenXML(url:String, dbHandler:MongoDBHandler):List
+getFileName(url:String):String
+checkSiteExn(url:String):boolean
+deleteXMLResourcesFolder(directory:File)

XMLFileReader

+findXMLFiles(folder:File):List
+checkExtension(file:File, extension:String):boolean

XMLReader

-dataMap:Map
-protocolMap:Map
-agendaMap:Map
-fractionMap:Map
-partyMap:Map
-speakerMap:Map
-speechMap:Map
-commentMap:Map
-speakerStammdatenMap:Map
-xmlFiles:List
-stammdatenFile:File
-dbHandler:MongoDBHandler

+XMLReader(xmlFiles:List, stammdatenFile:File, dbHandler:MongoDBHandler)
-analyseOneXML(file:File)
+readXML():Map
-checkXMLNotAnalysedUsingDB(file:File):boolean
-analyseStammdaten()
+getNodesFromXML(pNode:Node, sNodeName:String):List
+getSingleNodesFromXML(pNode:Node, sNodeName:String):Node
-normalizeFractionName(name:String, fraktionMap:Map):String
-normalizePartyName(name:String, partyMap:Map):String
+getOrmFiles():List

LaTeXHelper

LatexHelper

-dbHandler:MongoDBHandler
-converter:Converter

+LatexHelper()
+getTemplateAsString(templateName:String):String
+createLatexForAllProtocols()
+createLatexForSomeProtocols(startDate:long, endDate:long, collection:String)
+createPDF(latexCode:String)

InputException

+InputException()
+InputException(message:String)

Exception

ProgressBar

-progressName:String
-totalLength:int
-formatter:DecimalFormat

+ProgressBar(progressName:String,totalLength:int)
+showvalue:int)
-draw(rate:float, value:int)

NLPHelper

Node

-begin:int
-end:int
-value:String
-type:String
-pos:String
-score:double
-sentiment:double

+getBegin():int
+getEnd():int
+getValue():int
+getType():int
+getPos():int
+getScore():int
+getSentiment():int

NLPEngine

-builder:AggregateBuilder

+getBuilder():AggregateBuilder
+initPipeline():boolean

Converter

-engine:NLPEngine
-categoriesMapper:CategoriesMapper
-gson:Gson
-settings:JsonWriterSettings

+Converter()
+convertAll(downloadFrom:String, uploadTo:String, mongoDBHandler:MongoDBHandler)
+convertOne(toCollection:MongoCollection, doc:Document)
+analyzeSpeechDoc(speechDoc:Document):Document
+getNamedEntities(Cas:JCas):List
+getTokens(Cas:JCas):List
+getSentences(Cas:JCas):List
+getSentiment(Cas:JCas):double
+getLemma(Cas:JCas):List
+getDDCs(Cas:JCas):List
+getPOSs(Cas:JCas):List
+textToCAS(text:String):JCas
+serializeJCas(Cas:JCas):String
+deserializeJCas(serializedSpeechCAS:String):JCas
+speechNLPConstructor(doc:Document):Speech_NLP_Impl
+commentNLPConstructor(doc:Document):Comment_NLP_Impl
+protocolNLPConstructor(doc:Document):Protocol_NLP_Impl

CategoriesMapper

-categories:List

+CategoriesMapper()
+getCategories():List
+getCategory(search:String):String

javaspark

StartAPI

-dbHandlerAPI:MongoDBHandler
-configuration:Configuration

+main()
+init()
+getTopDocumentList(request:Request, subject:String, dataService:DataService, collection:String):List
+getTopNamedEntityList(request:Request, dataService:DataService, collection:String):List
+getTopSpeakerList(request:Request, dataService:DataService, collection:String):List
+getSentiments(request:Request, dataService:DataService, collection:String):List
+getUploadStatus(request:Request, response:Response, dbHandlerAPI:MongoDBHandler):JSONArray
+addNewData(request:Request, response:Response, dbHandlerAPI:MongoDBHandler):JSONArray
+applyCommentFilter(request:Request, dataService:DataService):JSONArray
+applyCategoryFilter(request:Request, dataService:DataService):JSONArray
+protocolFilterComment(request:Request, dataService:DataService):JSONArray
+protocolFilterCateg(request:Request, dataService:DataService):JSONArray
+protocolFilterSpeech(request:Request, dataService:DataService):JSONArray
+applySpeechFilter(request:Request, dataService:DataService):JSONArray

<<interface>>
SparkApplication

UserInterfaceConsole

+main()