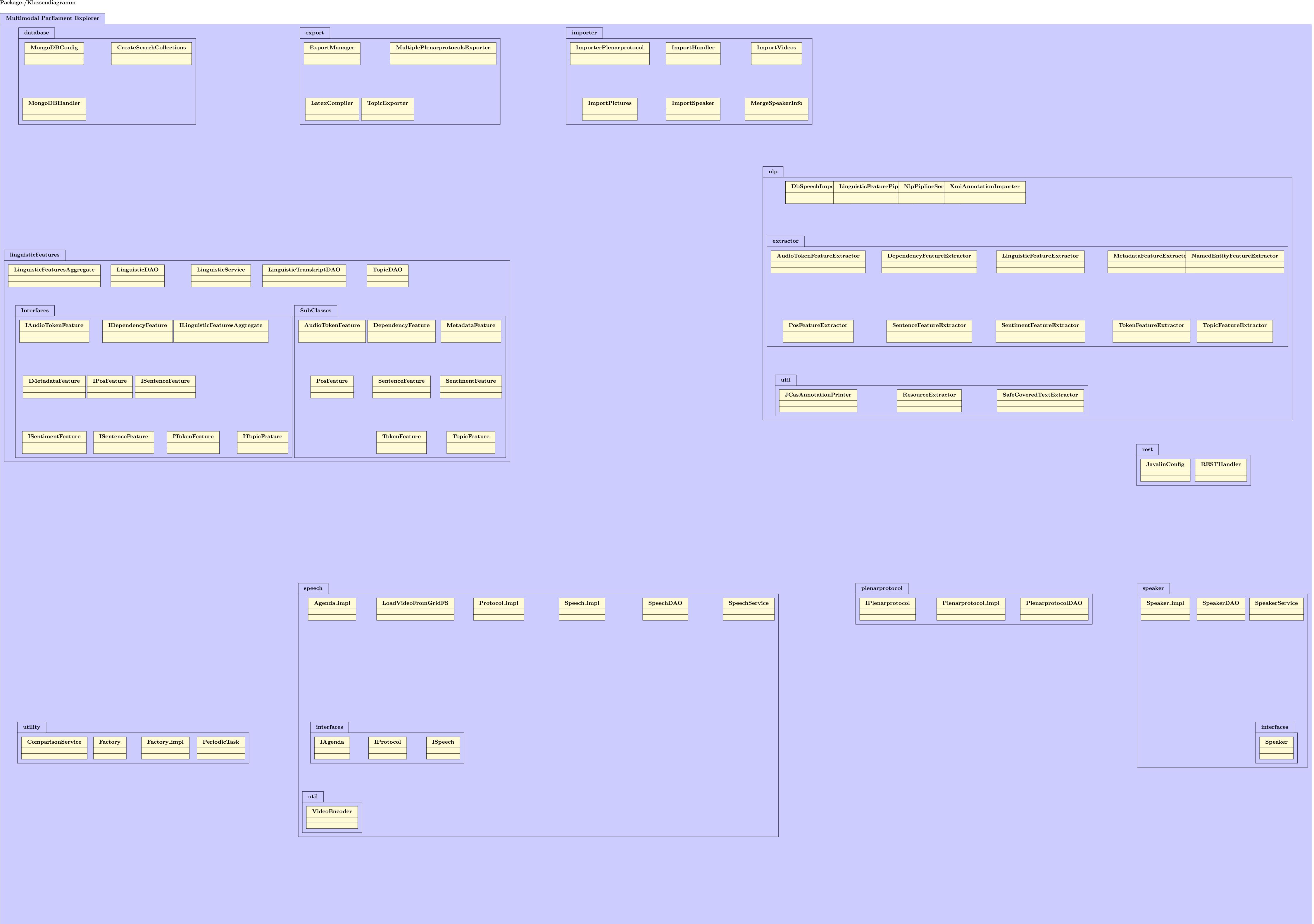
Klassen und Package Diagramm für Übung 5 PPR 24/25

Philipp Noah Hein, 6356965 March 21, 2025



database

MongoDBHandler

- instance : MongoDBHandler
 mongoClient : MongoClient
 database : MongoDatabase
 gridFSBucket : GridFSBucket
 config : MongoDBConfig
- + MongoDBHandler() : void
- + MongoDBHandler(boolean): void
- + getInstance() : MongoDBHandler
- + shutdown() : void
- + getConnectionString() : String
- + getDatabaseName() : String
- + connect() : MongoDatabase
- + close() : void
- + getPlenarprotocolsCollection() : MongoCollection<Document>
- + getSpeakerCollection() : MongoCollection<Document>
- + getSpeechCollection() : MongoCollection<Document>
- + getLinguisticFeaturesTranskriptCollection() : MongoCollection<Document>
- + getLinguisticFeaturesCollection() : MongoCollection<Document>
- + getTopicsCollection() : MongoCollection<Document>
- + getCollection(String) : MongoCollection<Document>
- + insertDocument(String, Document) : void
- + findDocuments(String, Bson) : List<Document>
- + findOne(String, Bson) : Document
- + findAll(String) : List<Document>
- + updateDocuments(String, Bson, Bson) : UpdateResult
- + deleteDocuments(String, Bson) : DeleteResult
- + aggregateDocuments(String, List<Bson>) : AggregateIterable<Document>
- + countDocuments(String, Bson) : long
- + getGridFSBucket() : GridFSBucket

CreateSearchCollections

- topicToRedeIds: Map;String, Set;String;
- topicDocuments : List;Document¿
- + main(String[]) : void

MongoDBConfig

- remotehost : String

remotedatabase : Stringremoteuser : String

- remotepassword : String

- remoteport : String

+ MongoDBConfig() : void

+ getURI() : String + getHost() : String

+ getDatabase() : String

+ getUser() : String + getPassword() : String

+ getPort() : String

export

ExportManager

- + clearExports(): void
- + getExportData(String) : byte[]
- + createExportInProgress(String): void
- + markFailedExport(String): void
- + createFinishedExport(String, byte[]): void
- + createFinishedExport(String, String): void
- + findExports(): Map<String, String>
- createExportDirIfNotExists(): void

LatexCompiler

- + compile(String) : Path
- + compileToByteArray(String): byte[]

MultiplePlenarprotocolsExporter

- + toTex(List<String>, boolean): String
- + toXML(List<String>): String
- + toPDF(List<String>, boolean): byte[]

TopicExporter

- + toTex(String, boolean): String
- + toPDF(String, boolean): byte[]
- + toXML(String) : String

importer

ImportHandler

- BASE_URL : String

- LIMIT_PARAM : String

- mongoDBHandler : MongoDBHandler

minIndex : intoffset : int

+ ImportHandler(int) : void

+ main(String[]) : void

+ importAllProtocols() : void

+ processProtocol(String) : int

+ extractProtocolIndex(String) : int

+ extractProtocolInfo(Document) : Document

+ extractAgendaInfo(Element) : Document

+ createSpeechDocument(Element, Document, Document): Document

+ extractSpeakerId(Element) : String

+ extractTextContent(Element, String) : List<Document>

+ extractNestedTextContent(Element, String, String, List<Document>): void

+ extractSpeechText(Element) : String

+ getCurrentSpeakerForElement(Element, String) : String

+ generateRandomId() : String

+ parseGermanDate(String) : long

+ parseTimeToMillis(String, long) : long

+ downloadContent(String) : String

ImportPictures

- BASE_URL : String

+ main(String[]) : void

+ createSpeakerMap(MongoCollection<Document>): Map<String, Document>

+ sendGetRequest(String) : String

+ sendPostRequest(String, String) : String

+ downloadImage(String) : byte[]

+ extractCsrfToken(String) : String

+ findMatchesAndUpdateCollection(String, Map<String, Document>, MongoCollection<Document>): int

+ extractNameFromContext(String) : String

+ findMatchingSpeaker(String, Map<String, Document>): Document

+ updateSpeechesWithSpeakerImageUrl(String, String, MongoCollection<Document>): int

+ updateSpeechesWithSpeakerImage(String, String, byte[], MongoCollection<Document>): int

ImportSpeaker

- xmlFile : File

- fis : FileInputStream

- dbFactory : DocumentBuilderFactory

- dBuilder : DocumentBuilder

- xmlDoc: org.w3c.dom.Document

- sdf : SimpleDateFormat

+ main(String[]) : void

- getTagValue(String, Element) : String

ImportVideos

- BASE_URL : String

- VIDEO_ID_PATTERN : Pattern

- DOWNLOAD_DIRECTORY : String

- SKIP_TITLES : Set<String>

+ main(String[]) : void

- splitSpeakerName(String) : String[]

- shouldSkipSpeaker(String, String): boolean

- extractVideoId(String) : String

- downloadFile(String) : byte[]

MergeSpeakerInfo

 $+ \min(String[]) : void$

linguisticFeatures

Interfaces

${f IAudio Token Feature}$

- begin: int - end : int

- coveredText : String - timeStart : double - timeEnd : double

+ get/set... : void

IDependency Feature |

- begin : int - end : int

- coveredText : String - dependencyType : String - governor : String - dependent : String

+ get/set... : void

ILinguisticFeaturesAggregate

- metadata : MetadataFeature_Impl - sentences : List<SentenceFeature_Impl> - tokens : List<TokenFeature_Impl> - sentiments : List<SentimentFeature_Impl> - namedEntities : List<NamedEntityFeature_Impl>

- topics : List<TopicFeature_Impl> - posCounts : Map<String, Integer> - namedEntityCounts : Map<String, Integer> - dependencyCounts : Map<String, Integer> - lemmaFrequency : Map<String, Integer>

- overallSentiment : double - sentimentDistribution : Map<String, Double> - topicCounts : Map<String, Integer> - audioTokens : List<AudioTokenFeature_Impl>

- posFeatures : List<PosFeature_Impl>

+ get/set... : void

- redeId : String

${f IMetadataFeature}$

- dependencies : List<DependencyFeature_Impl>

- language : String - documentId : String + get/set... : void

- documentTitle : String

- begin : int

- end : int - coveredText : String - pos : String - coarsePos : String

+ get/set... : void

IPosFeature

ISentence**F**eature - begin : int

- end : int - coveredText : String + get/set... : void

ISentimentFeature

- begin : int - end : int

- coveredText : String - sentiment : double - posScore : double

+ get/set... : void

- neuScore : double - negScore : double - subjectivity : double

SentimentFeature

ISentenceFeature

- begin : int

- end : int - coveredText : String + get/set... : void

- begin : int - end : int - coveredText : String - lemma : String - pos : String + get/set... : void

ITokenFeature

ITopicFeature

- begin : int - end : int - coveredText : String - value : String

- score : double + get/set... : void

SubClasses

AudioTokenFeature

- timeStart : double

- begin : int - end : int - coveredText : String

> - timeEnd : double + get/set... : void

DependencyFeature

- begin : int - end : int - coveredText : String - dependencyType : String

+ get/set... : void

- governor : String

- dependent : String

MetadataFeature - documentTitle : String

> - language : String + get/set... : void

- documentId : String

PosFeature

- begin : int - end : int - coveredText : String

- coarsePos : String + get/set... : void

- pos : String

- begin : int - end : int - coveredText : String

SentenceFeature

+ SentenceFeature_Impl(): void + SentenceFeature_Impl(int, int, String) : void + get/set... : void

- begin: int

- end : int - coveredText : String - sentiment : double - posScore : double

- neuScore : double - negScore : double - subjectivity : double

+ SentimentFeature_Impl(): void

+ SentimentFeature_Impl(int, int, String, double, double, double, double, double): void + get/set...: void

TokenFeature

- begin : int - end : int - coveredText : String - lemma : String

- pos : String

+ TokenFeature_Impl() : void

+ TokenFeature_Impl(int, int, String, String, String): void + get/set... : void

- begin : int - end : int - coveredText : String

- value : String - score : double

+ TopicFeature_Impl() : void

+ TopicFeature_Impl(int, int, String, String, double) : void

TopicFeature

+ get/set... : void

LinguisticFeaturesAggregate

- redeId : String

- metadata : MetadataFeature_Impl - sentences : List<SentenceFeature_Impl> - tokens : List<TokenFeature_Impl> - sentiments : List<SentimentFeature_Impl> - namedEntities : List<NamedEntityFeature_Impl> - dependencies : List<DependencyFeature_Impl>

- topics : List<TopicFeature_Impl> - audioTokens : List<AudioTokenFeature_Impl> - posFeatures : List<PosFeature_Impl> - posCounts : Map<String, Integer> - namedEntityCounts : Map<String, Integer>

- dependencyCounts : Map<String, Integer> - lemmaFrequency : Map<String, Integer> - overallSentiment : double

- sentimentDistribution : Map<String, Double> - topicCounts : Map<String, Integer>

- posCountsCoarse : Map<String, Integer> - topicsSearchField : String

+ get/set... : void+ removeOverallTextTopics() : void

+ calculateTopicsSearchField(): void + calculateStatistics() : void + calculatePOSCounts() : void

+ calculateNamedEntityCounts(): void + calculateOverallSentiment() : void + calculateLemmaFrequency() : void + calculateDependencyCounts() : void

+ calculateTopicCounts() : void

+ calculateSentimentDistribution() : void

LinguisticDAO

- collection : MongoCollection < Document > - dbHandler : MongoDBHandler - gson : Gson

+ LinguisticDAO(): void + save(LinguisticFeaturesAggregate_Impl) : void

+ delete(String) : void

+ findByDocumentId(String) : Optional<LinguisticFeaturesAggregate_Impl> + update(LinguisticFeaturesAggregate_Impl) : void

+ parseDocument(Document) : LinguisticFeaturesAggregate_Impl

+ getDocumentFromClass(LinguisticFeaturesAggregate_Impl) : Document

LinguisticService

- linguisticDAO : LinguisticDAO

+ LinguisticService(): void

+ createLinguisticFeatures(LinguisticFeaturesAggregate_Impl) : void + getLinguisticFeatures(String) : LinguisticFeaturesAggregate_Impl + updateLinguisticFeatures(String, LinguisticFeaturesAggregate_Impl) : void

+ deleteLinguisticFeatures(String) : void

- collection : MongoCollection < Document >

- dbHandler : MongoDBHandler - gson : Gson

+ LinguisticTranskriptDAO() : void + save(LinguisticFeaturesAggregate_Impl) : void

+ findByDocumentId(String) : Optional < LinguisticFeaturesAggregate_Impl> + update(LinguisticFeaturesAggregate_Impl) : void

LinguisticTranskriptDAO

+ delete(String) : void + close() : void

+ parseDocument(Document) : LinguisticFeaturesAggregate_Impl + getDocumentFromClass(LinguisticFeaturesAggregate_Impl) : Document

- collection : MongoCollection < Document > - dbHandler : MongoDBHandler

+ TopicDAO() : void

+ getAllTopics() : List<Document>

+ getSpeechesByTopic(String) : List<Speech_impl>

TopicDAO

+ getSpeechIdsByTopics(List<String>) : Set<String> + getSpeechesByTopics(List<String>) : List<Speech_impl>

+ save(String) : void

+ findByTopic(String) : String

+ update(String) : void

+ delete(String) : void + close() : void

extractor

AudioTokenFeatureExtractor	DependencyFeatureExtractor	LinguisticFeatureExtractor	Metada	ataFeatureExtractor	NamedEntityFeatureExtractor
+ extract(JCas) : List <iaudiotokenfeature></iaudiotokenfeature>	+ extract(JCas) : List <idependencyfeature></idependencyfeature>	+ extract(JCas) : List <t></t>	+ extract(JC	Cas) : MetadataFeature_Impl	+ extract(JCas) : List <inamedentityfeature></inamedentityfeature>

PosFeatureExtractor	SentenceFeatureExtractor	SentimentFeatureExtractor	TokenFeatureExtractor
+ extract(JCas) : List <iposfeature></iposfeature>	+ extract(JCas) : List <isentencefeature></isentencefeature>	+ extract(JCas) : List <isentimentfeature></isentimentfeature>	+ extract(JCas) : List <itokenfeature></itokenfeature>

TopicFeatureExtractor				
+ extract(JCas) : List <itopicfeature></itopicfeature>				

LinguisticFeaturePipline

- transkript DAO : Linguistic Transkript DAO
- textDao : LinguisticDAO
- $-\ linguistic Features Aggregate: Linguistic Features Aggregate_Impl$
- $+ \ Linguistic Feature Pipeline (Linguistic DAO, \ Linguistic Transkript DAO): void$ + processJCAS(JCas, TypeOfImport) : void

${\bf XmiAnnotation Importer}$

- threadLocalJCas : ThreadLocal<JCas>
- linguisticDAO : LinguisticDAO
- linguisticTranskriptDAO : LinguisticTranskriptDAO - executor : ExecutorService
- + XmiAnnotationImporter() : void
- + importAnnotations(String) : void
- + pushJCas(JCas, String, LinguisticDAO, LinguisticTranskriptDAO) : void + main(String[]) : void

NlpPiplineService

- composer : DUUIComposer
- + NlpPipelineService(DocumentType) : void
- + processCas(JCas) : void
- + getComposer() : DUUIComposer

util

JCasAnnotationPrinter

+ printAnnotationsFromAllViews(JCas, boolean) : void

${\bf Resource Extractor}$

- $+ \min(String[]) : void$
- extractResources(String, String) : void - extractFile(File, File) : void - removeGzExtension(String) : String

${\bf Safe Covered Text Extractor}$

- + getSafeCoveredText(Annotation, String) : String + getSafeEnd(Annotation, String): int
- THREAD_COUNT : int
- TIMEOUT_HOURS : int

- linguisticTranskriptDAO : LinguisticTranskriptDAO
- + DbSpeechImporter() : void
- + main(String[]) : void
- + importSpeeches(List<String>, boolean) : void
- createThreadPool() : ExecutorService
- fetchSpeeches(List<String>, ExecutorService): void
- fetchSpeechBatch(List<String>, AtomicInteger, int): void
- processSpeeches(ExecutorService, boolean) : void
- processTextComponent(JCas, NlpPipelineService, LinguisticFeaturePipeline, String) : void
- processVideoComponent(JCas, NlpPipelineService, String) : void
- isEmpty(String) : boolean
- shutdownExecutor(ExecutorService) : void
- printSummary(long, int) : void

${\bf DbSpeechImporter}$

- LOGGER : Logger
- BATCH_SIZE : int
- speechMap : Map<String, Speech_impl>mongoDbHandler : MongoDBHandler
- speechDAO : SpeechDAO
- linguisticDAO : LinguisticDAO
- $\hbox{-loadVideoFromGridFS}: LoadVideoFromGridFS$

- processSpeechBatch(List<Speech_impl>, AtomicInteger, int, boolean) : void
- $-\ process Single Speech (Speech_impl,\ NlpPipeline Service,\ NlpPipeline Service,\ NlpPipeline Service,\ Linguistic Feature Pipeline,\ AtomicInteger,\ int,\ boolean):\ void$
- $-\ process Transcript Component (JCas,\ NlpPipeline Service,\ Linguistic Feature Pipeline,\ String):\ void$

ріенагрі отокої

plenarprotocol

IPlenarprotocol

+ getId() : String
+ setId(String) : void
+ getProtocol() : IProtocol
+ setProtocol(IProtocol) : void
+ getSpeeches() : List<ISpeech>
+ setSpeeches(List<ISpeech>) : void
+ toTex(boolean) : String

${\bf Plenar protocol_impl}$

- _id : String

- protocol : IProtocol- speeches : List<ISpeech>

+ Plenarprotocol_impl(String, IProtocol, List<ISpeech>): void

+ getId() : String + setId(String) : void

+ getProtocol() : IProtocol + setProtocol(IProtocol) : void

+ getSpeeches() : List<ISpeech> + setSpeeches(List<ISpeech>) : void

+ toTex(boolean) : String + toPDF(boolean) : byte[]

+ toXML(): String

+ getAgendaSpeechesMap(): Map<IAgenda, List<ISpeech>>

PlenarprotocolDAO

- dbHandler : MongoDBHandler

+ PlenarprotocolDAO(MongoDBHandler): void

+ PlenarprotocolDAO(): void

+ getProtocolById(String) : Plenarprotocol_impl

+ getAllProtocols() : List<Plenarprotocol_impl>

+ getProtocolsAmount(): Long

+ getMaxProtocolIndex() : Integer

rest

JavalinConfig

- + JavalinConfig(String): void
- + getJavalinPort(): int
- + showBanner(): boolean

RESTHandler

- dbHandler : MongoDBHandler
- speechDAO : SpeechDAO
- linguisticDAO: LinguisticDAO
- plenarprotocolDAO : PlenarprotocolDAO
- topicDAO : TopicDAO
- + registerRoutes(Javalin) : void
- + showIndexPage(Context) : void
- + getExports(Context) : void
- + getExportData(Context) : void
- + clearExports(Context): void
- + exportSpeech(Context) : void
- + exportBySpeaker(Context): void
- + export(Context, boolean, String, String, Object, boolean): void
- + exportByTopic(Context) : void
- + exportByProtocol(Context) : void
- + exportMultipleProtocols(Context): void
- + getAllSpeeches(Context): void
- + getSpeechById(Context) : void
- + getLinguisticFeatures(Context): void
- + getLinguisticFeatureById(Context): void
- + showVisualizations(Context): void
- + getAllTopics(Context): void
- + getLinguisticFeaturesByTopic(Context): void
- + getAllSpeechesService(Context): void
- + showSpeechDetails(Context): void
- + getAllSpeechesPaginated(Context): void
- + getAllAvailableFactions(MongoCollection;Document;): List;String;
- + deleteSpeech(Context) : void
- + addComment(Context) : void
- + updateSpeech(Context) : void

 $\mathbf{speaker}$

interfaces

${\bf Speaker}$

- _id : String
- name : String - firstName : String
- title : String
- geburtsdatum : LocalDate
- geburtsort : String
- sterbedatum : LocalDate
- geschlecht : String
- beruf : String
- akademischertitel : String
- familienstand : String
- religion : String
- vita : String
- party : String
- memberships : Set¡Object¿
- imageUrl : String
- imageData : String
- + get/set... : void+ toTex(boolean) : String

- $Speaker_impl$
- _id : String
- name : String
- firstName : String
- title : String
- geburtsdatum : LocalDate
- geburtsort : String
- sterbedatum : LocalDate
- geschlecht : String
- beruf : String
- akademischertitel : String
- familienstand : String
- religion : String
- vita : String
- party : String
- memberships : Set¡Object¿
- imageUrl : String
- imageData : String
- + Speaker_impl(String, String, String) : void
- + Speaker_impl(String, String, + Speaker_impl(String, String, String, String, LocalDate, String, LocalDate, String, S
- + get/set... : void
- + convertStringToDate(String) : LocalDate
- + toTex(boolean) : String
- + toXML() : String
- + toPDF(boolean) : byte[]

${\bf SpeakerDAO}$

- speakerCollection : MongoCollection<Document>
- + SpeakerDAO(MongoCollection<Document>): void
- + findById(String) : Speaker_impl
- + loadAll() : List<Speaker_impl>

SpeakerService

 \mathbf{speech}

interfaces

 ${f IAgenda}$

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

+ get/set... : void

IProtocol

- date : Long

- starttime : Long - endtime : Long

- index : Integer

- title : String - place : String - wp : Integer

+ get/set... : void

ISpeech

- _id : String

- text : String

- speaker : String

- protocol : IProtocol

- textContent : ArrayList¡Object¿

- agenda : IAgenda - speakerObject : Speaker

+ get/set... : void+ toTex(boolean) : String util

 ${f VideoEncoder}$

+ encodeVideoToBase64(File) : String $+ \min(String[]) : void$

 ${\bf SpeechDAO}$

- speechCollection : MongoCollection < Document >

+ SpeechDAO(MongoCollection<Document>): void

+ SpeechDAO(): void

+ findBySpeakerId(String) : List<Speech_impl>

+ countBySpeakerId(String) : long + findById(String) : Speech_impl

+ findAll() : List<Speech_impl>

+ findAllPaginated(int, int, Set<String>, String, String) : List<Speech_impl>

+ count(Set<String>, String) : long

+ delete(String) : boolean

+ update(String, Speech_impl) : boolean

SpeechService

- speechDAO : SpeechDAO

+ SpeechService(SpeechDAO) : void

+ getAllSpeeches() : List<Speech_impl>

+ getSpeechById(String) : Speech_impl

+ getAllSpeechesPaginated(int, int, String, List<String>, String) : List<Speech_impl>

+ countSpeeches(String, List<String>): long

$\mathbf{Agenda_impl}$

- index : String - id : String

- title : String

+ Agenda_impl(String, String, String) : void + get/set... : void

 ${\bf Load Video From Grid FS}$

- mongoDbHandler : MongoDBHandler - gridFSBucket : GridFSBucket - speechDAO : SpeechDAO

+ LoadVideoFromGridFS(MongoDBHandler) : void

+ LoadVideoFromGridFS() : void

+ loadVideo(ObjectId) : byte[]+ getVideoBase64(Speech_impl) : HashMap<String, String>

+ transferToOutputStream(InputStream, OutputStream) : void

+ getSpeechDAO() : SpeechDAO $+ \min(String[]) : void$

- date : Long - starttime : Long

- endtime : Long

- index : Integer

- title : String

- place : String

- wp : Integer

+ Protocol_impl(Long, Long, Long, Integer, String, String, Integer): void

 ${
m Protocol_impl}$

+ get/set... : void

+ dateFormat(Long) : String

 ${f Speech_impl}$

- _id : String - text : String

- speaker : String

- protocol : IProtocol

- textContent : ArrayList¡Object¿

- agenda : IAgenda

- speakerObject : Speaker - videoUrl : String

- videoId : String

- videoFileId : ObjectId

+ Speech_impl(String, String, String, IProtocol, ArrayList;Object;, IAgenda, Speaker, String): void

+ get/set... : void

+ getVideoId() : String

+ setVideoId(String) : void + getVideoUrl() : String

+ setVideoUrl(String) : void

+ getVideoFileId() : ObjectId

+ setVideoFileId(ObjectId) : void

+ toCAS(LoadVideoFromGridFS, boolean) : JCas

+ toTex(boolean) : String

+ toXML() : String + toPDF(boolean) : byte[]

+ toDocument() : Document

utility

ComparisonService

- dbHandler : MongoDBHandler
- speechCollection : MongoCollection < Document >
- linguisticCollection : MongoCollection<Document>- linguisticTranskriptCollection : MongoCollection<Document>
- + ComparisonService() : void
- + findMissingLinguisticTranskriptFeatureIds() : List<String>
- + findMissingLinguisticTranskriptFeatureIds(int, boolean) : List<String>
- + findMissingLinguisticFeatureIds(int, boolean) : List<String>
- + findMissingLinguisticFeatureIds() : List<String>
- + getAllMissingLinguisticFeatureIds() : List<String>
- + getAllMissingLinguisticFeatureIds(int, boolean) : List<String>
- + close() : void

Factory

- + createSpeaker(Document) : Speaker_impl + createSpeech(Document) : Speech_impl
- + createProtocol(Document) : IProtocol
- + createAgenda(Document) : Agenda_impl
- + createPlenarprotocol(Document) : Plenarprotocol_impl

${f Factory_impl}$

- + createSpeaker(Document) : Speaker_impl
- + createSpeech(Document) : Speech_impl
- + createProtocol(Document) : Protocol_impl
- + createAgenda(Document) : Agenda_impl
- + createPlenarprotocol(Document) : Plenarprotocol_impl

${f Periodic Task}$

- scheduler : ScheduledExecutorService

+ startScheduledTask() : void

