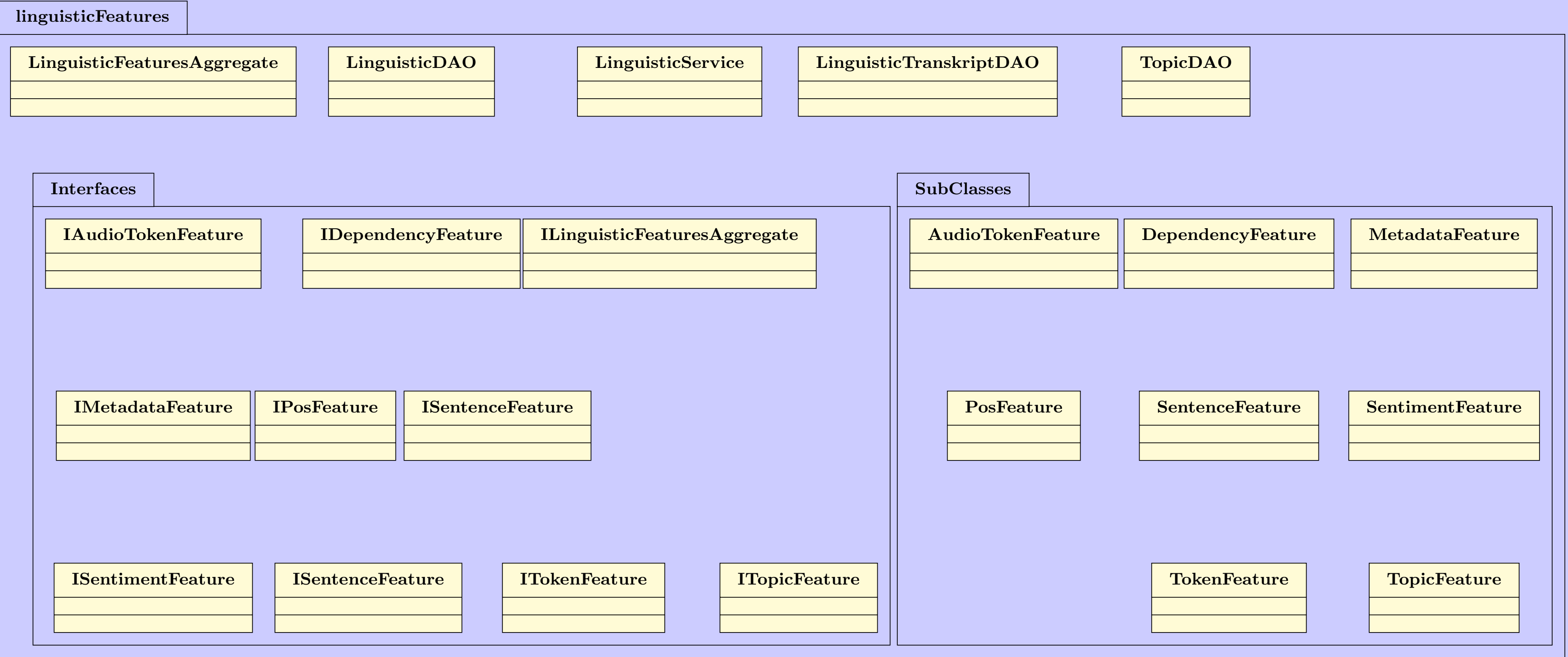
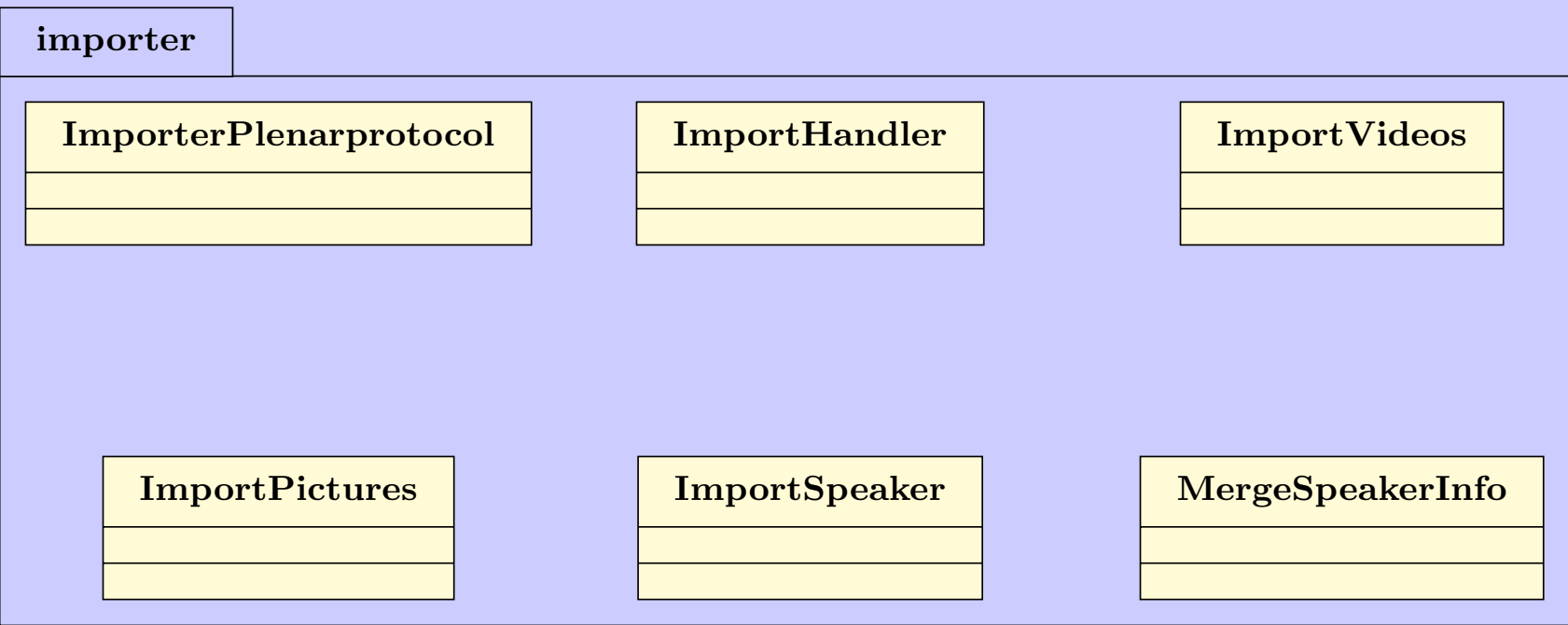
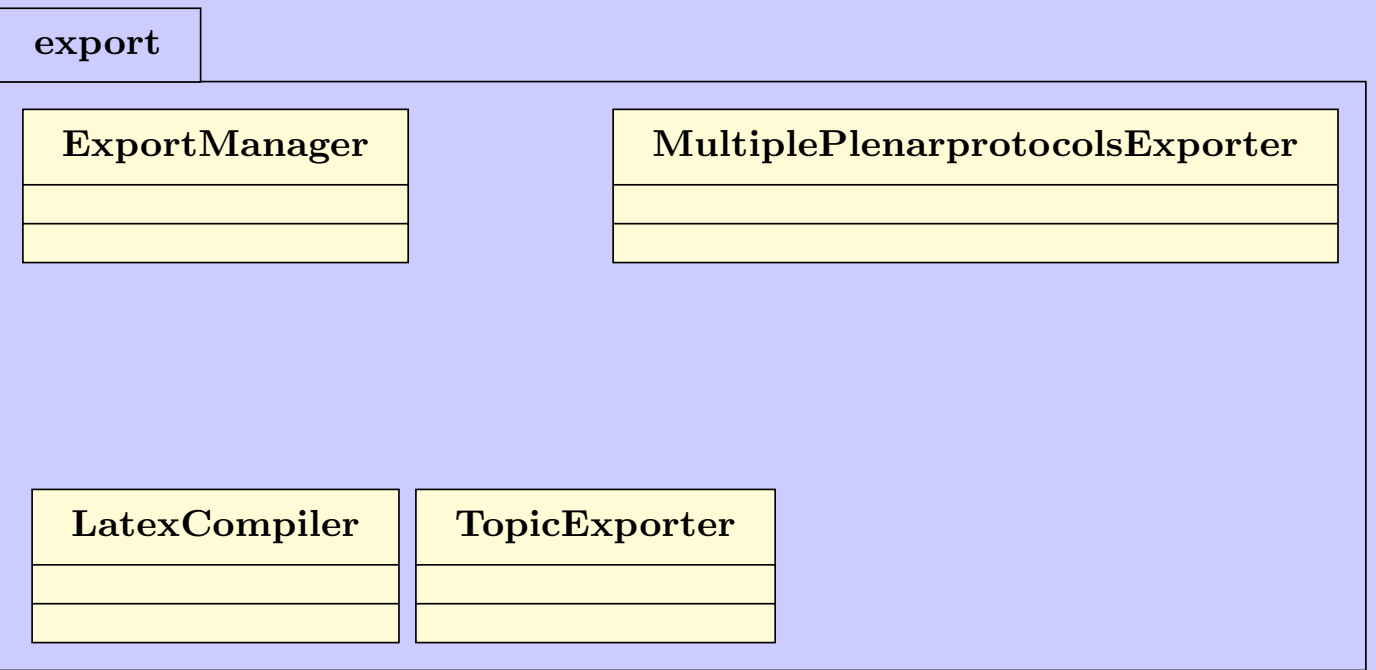
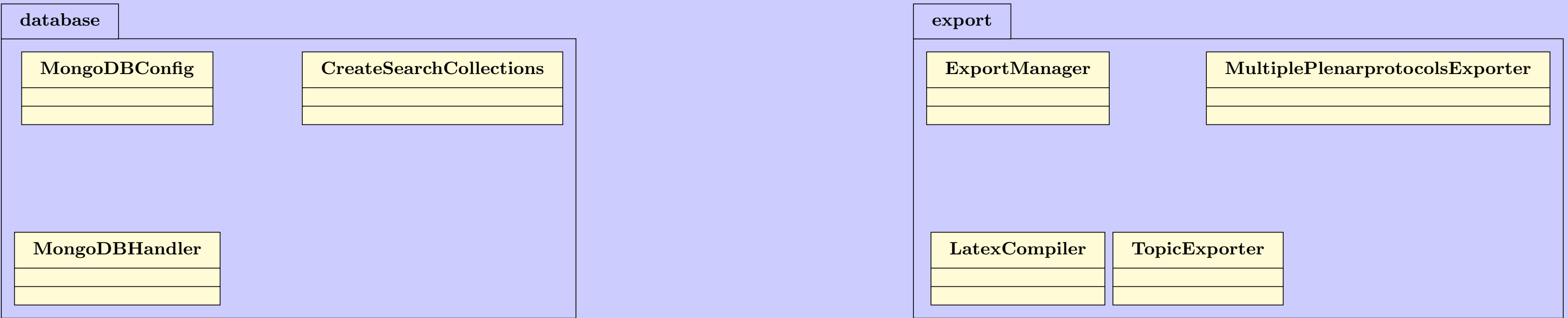


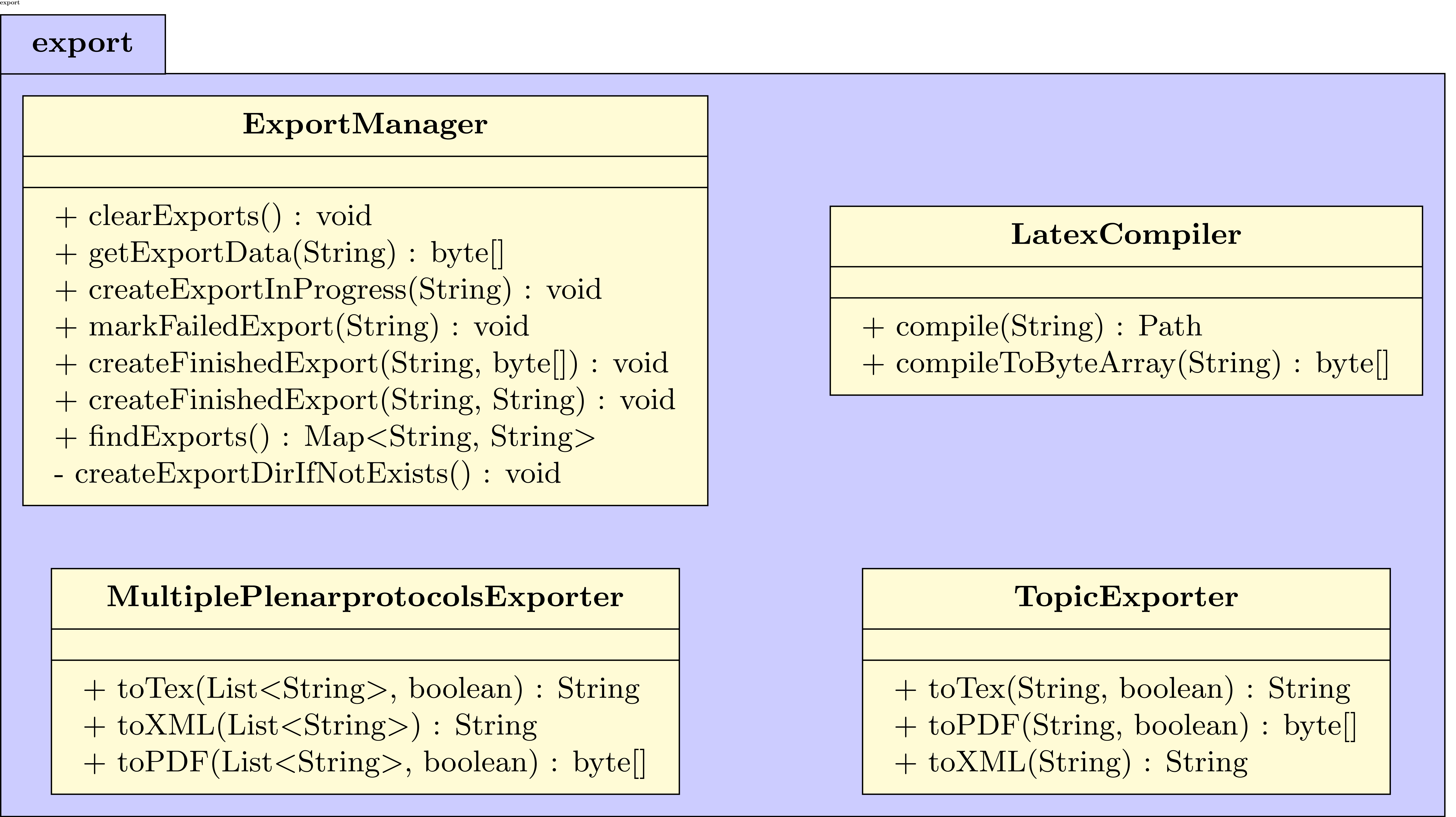


Multimodal Parliament Explorer









importer

ImportHandler

- BASE\_URL : String  
- LIMIT\_PARAM : String  
- mongoDBHandler : MongoDBHandler  
- minIndex : int  
- offset : 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

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[]

ImportSpeaker

- xmlFile : File  
- fis : FileInputStream  
- dbFactory : DocumentBuilderFactory  
- dBuilder : DocumentBuilder  
- xmlDoc : org.w3c.dom.Document  
- sdf : SimpleDateFormat

+ main(String[]) : void  
- getTagValue(String, Element) : String

MergeSpeakerInfo

+ main(String[]) : void



## linguisticFeatures

## Interfaces

SubClasses

AudioTokenFeature

- begin : int  
- end : int  
- coveredText : String  
- timeStart : double  
- timeEnd : double  
+ get/set... : void

DependencyFeature

- begin : int  
- end : int  
- coveredText : String  
- dependencyType : String  
- governor : String  
- dependent : String  
+ get/set... : void

MetadataFeature

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

PosFeature

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

SentenceFeature

- begin : int  
- end : int  
- coveredText : String  
+ SentenceFeature\_Impl() : void  
+ SentenceFeature\_Impl(int, int, String) : void  
+ get/set... : void

SentimentFeature

- 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

TopicFeature

- begin : int  
- end : int  
- coveredText : String  
- value : String  
- score : double  
+ TopicFeature\_Impl() : void  
+ TopicFeature\_Impl(int, int, String, String, double) : void  
+ get/set... : void

## SubClasses

```

classDiagram
    class AudioTokenFeature {
        - begin : int
        - end : int
        - coveredText : String
        - timeStart : double
        - timeEnd : double
        + get/set... : void
    }
    class DependencyFeature {
        - begin : int
        - end : int
        - coveredText : String
        - dependencyType : String
        - governor : String
        - dependent : String
        + get/set... : void
    }
    class MetadataFeature {
        - documentTitle : String
        - language : String
        - documentId : String
        + get/set... : void
    }
    class PosFeature {
        - begin : int
        - end : int
        - coveredText : String
        - pos : String
        - coarsePos : String
        + get/set... : void
    }
    class SentenceFeature {
        - begin : int
        - end : int
        - coveredText : String
        + SentenceFeature_Impl() : void
        + SentenceFeature_Impl(int, int, String) : void
        + get/set... : void
    }
    class SentimentFeature {
        - begin : int
        - end : int
        - coveredText : String
        - 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
    }
    class 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
    }
    class TopicFeature {
        - begin : int
        - end : int
        - coveredText : String
        - value : String
        - score : double
        + TopicFeature_Impl() : void
        + TopicFeature_Impl(int, int, String, String, double) : void
        + get/set... : void
    }
    SentimentFeature <|-- TokenFeature
    SentimentFeature <|-- TopicFeature
    SentenceFeature <|-- MetadataFeature
    SentenceFeature <|-- PosFeature
    SentenceFeature <|-- DependencyFeature
    AudioTokenFeature <|-- TokenFeature
  
```

The diagram illustrates the class hierarchy and attributes for various NLP features. The classes are organized into a hierarchy where **SentimentFeature** is the base class for **TokenFeature** and **TopicFeature**. **SentenceFeature** is the base class for **MetadataFeature** and **PosFeature**. **DependencyFeature** is a subclass of **SentenceFeature**. **AudioTokenFeature** is a subclass of **TokenFeature**.

**AudioTokenFeature** attributes:

- begin : int
- end : int
- coveredText : String
- timeStart : double
- timeEnd : double
- + get/set... : void

**DependencyFeature** attributes:

- begin : int
- end : int
- coveredText : String
- dependencyType : String
- governor : String
- dependent : String
- + get/set... : void

**MetadataFeature** attributes:

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

**PosFeature** attributes:

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

**SentenceFeature** attributes:

- begin : int
- end : int
- coveredText : String
- + SentenceFeature\_Impl() : void
- + SentenceFeature\_Impl(int, int, String) : void
- + get/set... : void

**SentimentFeature** attributes:

- begin : int
- end : int
- coveredText : String
- 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** attributes:

- begin : int
- end : int
- coveredText : String
- lemma : String
- pos : String
- + TokenFeature\_Impl() : void
- + TokenFeature\_Impl(int, int, String, String, String) : void
- + get/set... : void

**TopicFeature** attributes:

- begin : int
- end : int
- coveredText : String
- value : String
- score : double
- + TopicFeature\_Impl() : void
- + TopicFeature\_Impl(int, int, String, String, double) : void
- + get/set... : void

## AudioTokenFeature

<ul style="list-style-type: none"><li>- begin : int</li><li>- end : int</li><li>- coveredText : String</li><li>- timeStart : double</li><li>- timeEnd : double</li></ul>
+ get/set... : void

## DependencyFeature

<ul style="list-style-type: none"> <li>- begin : int</li> <li>- end : int</li> <li>- coveredText : String</li> <li>- dependencyType : String</li> <li>- governor : String</li> <li>- dependent : String</li> </ul>	
+ get/set... : void	

## MetadataFeature

<ul style="list-style-type: none"> <li>- documentTitle : String</li> <li>- language : String</li> <li>- documentId : String</li> </ul>
<ul style="list-style-type: none"> <li>+ get/set... : void</li> </ul>

## PosFeature

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

+ get/set... : void
```

## SentenceFeature

- begin : int - end : int - coveredText : String
+ SentenceFeature_Impl() : void + SentenceFeature_Impl(int, int, String) : void + get/set... : void

## SentimentFeature

```
- begin : int
- end : int
- coveredText : String
- sentiment : double
- posScore : double
- neuScore : double
- negScore : double
- subjectivity : double

+ SentimentFeatureImpl() : void
+ SentimentFeatureImpl(int, int, String, 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
```

## TonicFeature

```

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

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

```

## LinguisticFeaturesAggregate

```

- 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
+ calculateSentimentDistribution() : void
+ calculateTopicCounts() : void

```

## LinguisticDAC

```

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

+ LinguisticDAO() : void
+ save(LinguisticFeaturesAggregate_Impl) : void
+ findById(DocumentId(String) : Optional<LinguisticFeaturesAggregate_Impl>
+ update(LinguisticFeaturesAggregate_Impl) : void
+ delete(String) : 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

## LinguisticTranskriptDAO

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

---

```
+ LinguisticTranskriptDAO() : void
+ save(LinguisticFeaturesAggregate_Impl) : void
+ findById(DocumentId(String) : Optional<LinguisticFeaturesAggregate_Impl>
+ update(LinguisticFeaturesAggregate_Impl) : void
+ delete(String) : void
+ close() : void
+ parseDocument(Document) : LinguisticFeaturesAggregate_Impl
+ getDocumentFromClass(LinguisticFeaturesAggregate_Impl) : Document
```

## TopicDAO

```

- collection : MongoCollection<Document>
- dbHandler : MongoDBHandler

+ TopicDAO() : void
+ getAllTopics() : List<Document>
+ getSpeechesByTopic(String) : List<Speech_impl>
+ 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

+ extract(JCas) : List<IAudioTokenFeature>

DependencyFeatureExtractor

+ extract(JCas) : List<IDependencyFeature>

LinguisticFeatureExtractor

+ extract(JCas) : List<T>

MetadataFeatureExtractor

+ extract(JCas) : MetadataFeature\_Impl

NamedEntityFeatureExtractor

+ extract(JCas) : List<INamedEntityFeature>

PosFeatureExtractor

+ extract(JCas) : List<IPosFeature>

SentenceFeatureExtractor

+ extract(JCas) : List<ISentenceFeature>

SentimentFeatureExtractor

+ extract(JCas) : List<ISentimentFeature>

TokenFeatureExtractor

+ extract(JCas) : List<ITokenFeature>

TopicFeatureExtractor

+ extract(JCas) : List<ITopicFeature>

LinguisticFeaturePipeline

- transkriptDAO : LinguisticTranskriptDAO  
- textDao : LinguisticDAO  
- linguisticFeaturesAggregate : LinguisticFeaturesAggregate\_Impl

+ LinguisticFeaturePipeline(LinguisticDAO, LinguisticTranskriptDAO) : void  
+ processJCAS(JCas, TypeOfImport) : void

XmiAnnotationImporter

- threadLocalJCas : ThreadLocal<JCas>  
- linguisticDAO : LinguisticDAO  
- linguisticTranskriptDAO : LinguisticTranskriptDAO  
- executor : ExecutorService

+ XmiAnnotationImporter() : void  
+ importAnnotations(String) : void  
+ pushJCas(JCas, String, LinguisticDAO, LinguisticTranskriptDAO) : void  
+ main(String[]) : void

NlpPipelineService

- composer : DUUIComposer

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

JCasAnnotationPrinter

+ printAnnotationsFromAllViews(JCas, boolean) : void

ResourceExtractor

+ main(String[]) : void  
- extractResources(String, String) : void  
- extractFile(File, File) : void  
- removeGzExtension(String) : String

SafeCoveredTextExtractor

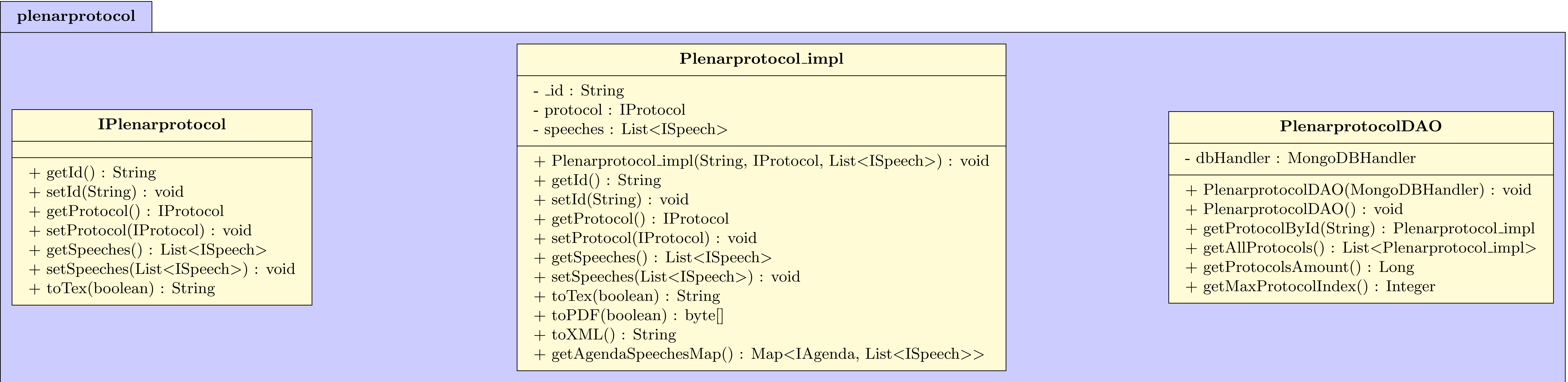
+ getSafeCoveredText(Annotation, String) : String  
+ getSafeEnd(Annotation, String) : int

DbSpeechImporter

- LOGGER : Logger  
- THREAD\_COUNT : int  
- BATCH\_SIZE : int  
- TIMEOUT\_HOURS : int  
- speechMap : Map<String, Speech\_impl>  
- mongoDbHandler : MongoDBHandler  
- speechDAO : SpeechDAO  
- linguisticDAO : LinguisticDAO  
- linguisticTranskriptDAO : LinguisticTranskriptDAO  
- loadVideoFromGridFS : LoadVideoFromGridFS

+ 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  
- processSpeechBatch(List<Speech\_impl>, AtomicInteger, int, boolean) : void  
- processSingleSpeech(Speech\_impl, NlpPipelineService, NlpPipelineService, NlpPipelineService, LinguisticFeaturePipeline, AtomicInteger, int, boolean) : void  
- processTextComponent(JCas, NlpPipelineService, LinguisticFeaturePipeline, String) : void  
- processVideoComponent(JCas, NlpPipelineService, String) : void  
- processTranscriptComponent(JCas, NlpPipelineService, LinguisticFeaturePipeline, String) : void  
- isEmpty(String) : boolean  
- shutdownExecutor(ExecutorService) : void  
- printSummary(long, int) : void

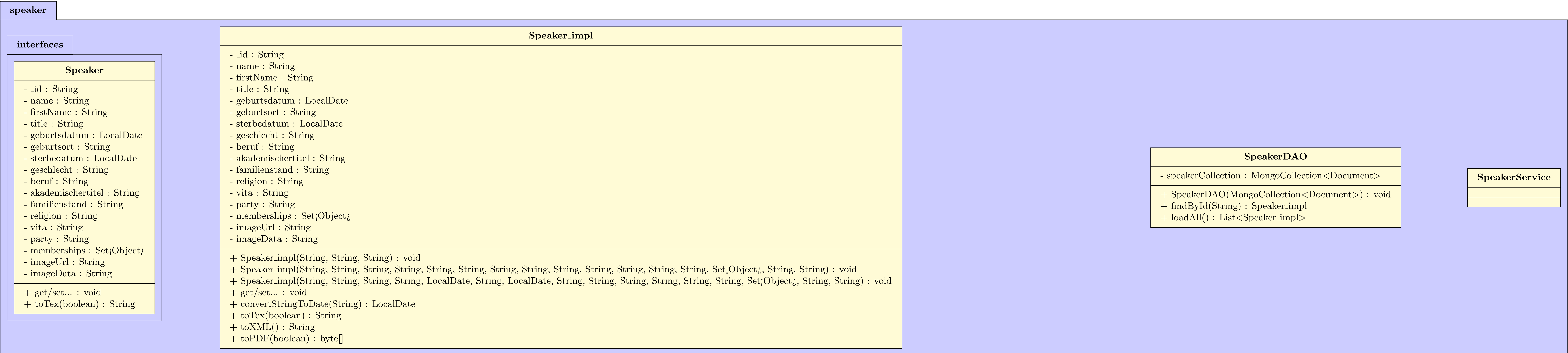




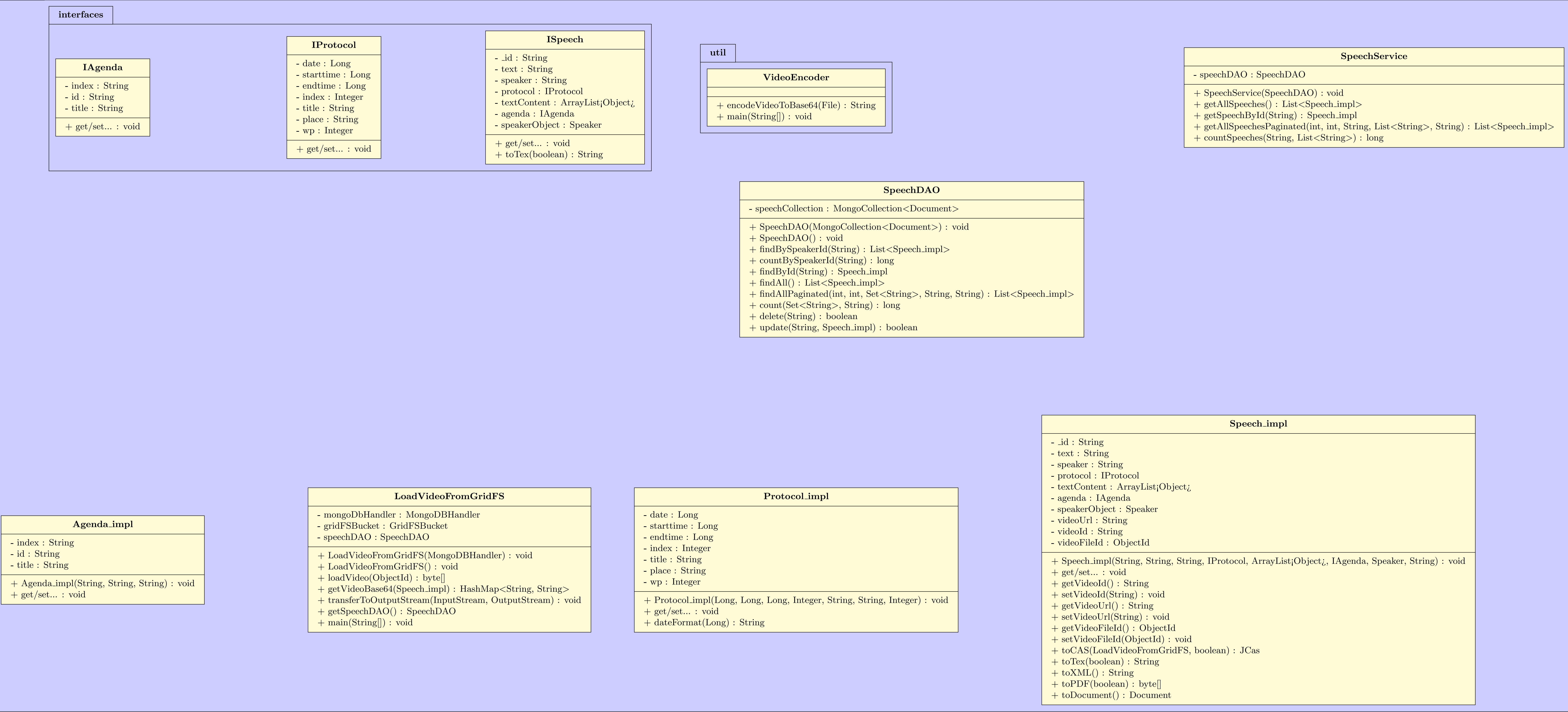


JavalinConfig
+ JavalinConfig(String) : void + getJavalinPort() : int + showBanner() : boolean

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







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

Factory
<div>+ createSpeaker(Document) : Speaker_impl</div> <div>+ createSpeech(Document) : Speech_impl</div> <div>+ createProtocol(Document) : IProtocol</div> <div>+ createAgenda(Document) : Agenda_impl</div> <div>+ createPlenarprotocol(Document) : Plenarprotocol_impl</div>

Factory_impl
<div>+ createSpeaker(Document) : Speaker_impl</div> <div>+ createSpeech(Document) : Speech_impl</div> <div>+ createProtocol(Document) : Protocol_impl</div> <div>+ createAgenda(Document) : Agenda_impl</div> <div>+ createPlenarprotocol(Document) : Plenarprotocol_impl</div>

PeriodicTask
<div>- scheduler : ScheduledExecutorService</div>
<div>+ startScheduledTask() : void</div>



