|  |  |
| --- | --- |
|  | **ETCS@EBD** |
|  | TMS Simulator und SL  Entwicklerdokumentation |
|  |  |
|  |  |
|  | Dokumentation, v0.4  Status: Draft, Stand: 21.10.2020 15:28  Autor: Werner Iberl |
|  |  |



|  |  |  |  |
| --- | --- | --- | --- |
|  | **Bearbeitung Editorial** | **Qualitätsprüfung Reviewer** | **Freigabe Approval** |
| **Name** | Werner Iberl | Georg Bolz | Frederik Düpmeier |
| **Position/Rolle** | Softwareentwickler | Projektleiter | Projektleiter |
| **Datum** | 21.10.20 |  |  |
| **Unterschrift** | gez. Iberl | gez. Bolz | gez. Düpmeier |

Versionshistorie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vers.** | **Datum** | **Kap.** | **Beschreibung** | **Autor** |
| **0.4** | **21/10/20** |  | **Smart Logic Komponenten Struktur** | **WI** |
| 0.3 | 17/08/20 |  | Javadoc für die SmartLogic eingepflegt | WI |
| 0.2 | 15/01/20 |  | Strukturierung des Programms | WI |
| 0.1 | 09/01/20 |  | Erster Entwurf | WI |

1 Beschreibung des TMS Simulators 8

2 Beschreibung der SmartLogic 9

2.1 Komponenten der smartLogic 10

3 Struktur und Funktionalität 12

3.1 Die SmartLogic 13

3.1.1 Class SmartLogic 13

3.1.2 Class EventBusManager 16

3.1.3 Enum BlockedArea.BLOCK\_Q\_SCALE 18

3.1.4 Class BlockedArea 19

3.1.5 Class QueueUuidMapper 20

3.1.6 Class PriorityMessage 22

3.1.7 Class RbcModul 23

3.1.8 Class SmartLogicTmsProxy 32

3.1.9 Class SmartServer 34

3.1.10 Class SmartServer4TmsImpl 35

3.1.11 Class TmsOuputWorker<T> 37

3.1.12 Class MaRequestReturnPayload 38

3.1.13 Class SmartServerMessage 39

3.1.14 Class SmartSafety 40

3.1.15 Class SmartSafetyContinousConnectTest 44

3.2 Der TMS-Simulator 45

3.2.1 Class MainTmsSim 45

3.2.2 Enum MainTmsSim.TmsMode 46

3.2.3 Class GraphicMoveByMouse 47

3.2.4 Class LoggingFrame 48

3.2.5 Class CartesianDialog 49

3.2.6 Class CartesianPanel 50

3.2.7 Class PositionReportController 52

3.2.8 Class SmartClient 54

3.2.9 Class SmartClientHandler 55

3.2.10 Class TmsMessage 57

3.2.11 Class TmsMovementAuthority 58

3.2.12 Class CheckMovementAuthority 59

3.2.13 Class Commands 60

3.2.14 Class CrossingSwitch 61

3.2.15 Enum PlanProTmsAdapter.PlanProVersion 62

3.2.16 Class TopologyGraph 63

3.2.17 Class TopologyFactory 69

3.2.18 Class BranchingSwitch 72

3.2.19 Class CrossoverModel 77

3.2.20 Class Rail 78

3.2.21 Class UiTools 80

3.2.22 Class CrossoverEnumModel 81

3.2.23 Class CrossoverMainModel 82

3.2.24 Class PlanData 83

3.2.25 Class SegmentAddController 85

3.2.26 Class CartesianSpeedModel 86

3.2.27 Class SpeedSegmentViewModel 87

3.2.28 Class AddSegmentDialog 88

3.2.29 Class SpeedDialog 89

3.2.30 Class SpeedPanel 90

3.2.31 Class EnumModel 91

3.2.32 Class MaCreatingFrame 93

3.2.33 Class TrackplanGraphicPanel 94

3.2.34 Class CrossoverController 95

3.2.35 Class MoveController 96

3.2.36 Class RouteController 97

3.2.37 Class TrackController 98

3.2.38 Class ZoomController 100

3.2.39 Class LinearLocationWaypointSetWindow 101

3.2.40 Class MainGraphicPanel 102

3.2.41 Class RouteComponent 105

3.2.42 Class RouteViewPort 108

3.2.43 Class SingleEnumSelectorComponent<T extends EnumModel> 110

3.2.44 Class TrackPanel 111

3.2.45 Class TrackWindow 112

3.2.46 Class WaypointDecorator 113

3.2.47 Class WaypointEnd 114

3.2.48 Class WaypointStart 115

3.2.49 Class ZoomFrame 116

3.2.50 Class ZoomPanel 117

3.2.51 Enum LinksRechts 118

3.2.52 Class TranslationModel 119

3.2.53 Class ZoomModel 121

3.2.54 Class TrainController 122

3.2.55 Class TrainDistance 124

3.2.56 Class TrainModel 125

3.2.57 Class SingleTrainSubPanel 127

3.2.58 Class TrainPanel 128

3.2.59 Class DefaultRepo<K,​V> 129

3.2.60 Class DoubleCoord 130

3.2.61 Class StatusMessageAppender 131

3.2.62 Class ThreadedRepo<K,​V> 132

3.2.63 Class UtilFunction 133

3.2.64 Class ConfigHandler 134

3.3 Die Generischen Klassen aus Planpro 135

[Abbildung 1 Komponenten der smartLogic 10](#_Toc54190011)

[Abbildung 2 Kommikation der beteiligten Module 24](#_Toc54190012)

# Beschreibung des TMS Simulators

Der TMS Simulator hat eine Oberfläche die ab Januar 2020 auf das PlanPro-Format fußt. Dieses Format bringt den Gleisplan in sich mit ein. Es können Stellwerkszustände des Gleisplanes geändert werden. Es können Fahrzeuge angelegt werden. Diese erhalten ein Laufweg mit Farbe und Eigenschaft wie Fahrzeugtyp über eine Oberfläche. Es soll außerdem ein Geschwindigkeitsprofil für die Strecke erstellt werden können. Die SmartLogic überprüft, ob die Fahrerlaubnis zulässig ist. Es wurde Konfigurierung durch einen eigenen Config-Handler im Modul ermöglicht. Diese verweist auf eine SL-Config. Es gibt einen Logging-Frame, der Weichenänderung als Log-Ausgabe darstellt. Dieser soll aber bald ersetzt werden, weil die Weichenlage im Plan ersichtlich werden sein soll.

# Beschreibung der SmartLogic

Die SmartLogic überprüft, ob an sie gestellte Movement Authorities gültig sind, bevor die Fahrstraße gestellt wird und das RBC eine Fahrerlaubnis für einen Zug erhält. Die wesentliche Prüfinstanz heißt SmartSafety und ist ein Untermodul. Es führt auch einen Selbstcheck auf sich selbst aus. Hierfür werden Unit-Tests innerhalb des laufenden Moduls ausgeführt. Die Smart-Logic besitzt einen EventBus. Er ermöglich es Log-Nachrichten zu übertragen. Die Smart-Logic bestitzt einen Gui-Server. Er kann die Nachrichten zu Gui-Clients absetzen.

Es folgt ein grober Aufweis der Komponenten innerhalb der smartLogc.

## Komponenten der smartLogic

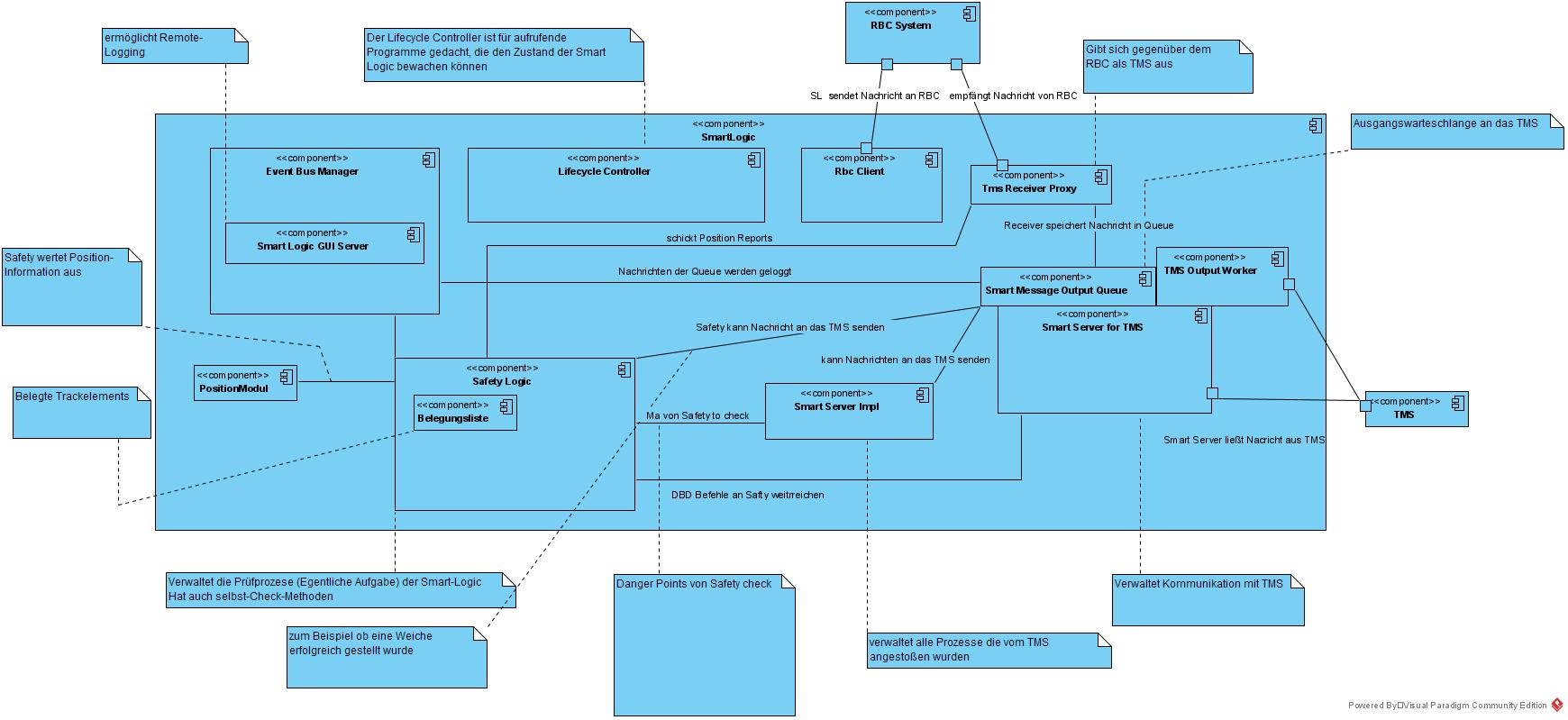


Abbildung 1 Komponenten der smartLogic

Die Abbildung 1 der Komponenten wird im Folgenden im Uhrzeigersinn durchgegangen. Zuerst wird oben auf die Schnittstelle zum RBC eingegangen. Innerhalb der smartLogic dient der RBC-Client dazu, Nachrichten an das RBC (RBC-System in Abbildung 1) zu senden. Der RBC-Client sendet ausschließlich. Der TMS-Receiver Proxy ist für den Empfang zuständig. Die eingehenden Nachrichten werden von Safety-Logic (unten links) ausgewertet. Die Safety-Logic ist die eigentliche Komponente, in der Sicherungsfunktionen durchgeführt werden. Der TMS Receiver Proxy speichert die Nachricht vom RBC in die Ausgangswarteschlange für das TMS (Smart Message Output Queue). Es läuft ständig ein Worker-Modul, dass Nachrichten aus der Warteschlange an das TMS weiterleitet. Die smartLogic verfügt über einen Smart Server für das TMS, der Nachrichten aus dem TMS verarbeitet und gewünschte Prozesse der smartLogic anstößt. Diese Prozesse werden in der Smart Server Implementierung (Smart Server Impl genannt) verwaltet. Sicherheitsrelevante Anfragen werden von dort an die Safety-Logic weitergereicht. Intern werden belegte Streckenabschnitte in einer Belegungsliste verwaltet. Die smartLogic empfängt aus mehreren Quellen Positionsangaben. Alle Informationen werden im Positions-Modul gehalten. Dieses Modul kann man sich wie eine Datenbank im Speicher vorstellen.

Zusätzlich zur Verwaltung der smartLogic gibt es einen Event Bus Manager. Er leitet Log-Daten an Clients weiter. Den Zustand der smartLogic ist von Aufruf-Programmen überwachbar. Man kann dafür den Lifecycle-Controller von außen aufrufen.

# Struktur und Funktionalität

Das Packet von TMS und SmartLogic greifen auf die gleichen Klassen zurück.

Es ändert sich nur der Main-Entry-Point des Moduls, das dann zweimal gestartet werden kann.

Der Main-Entry-Point des TMS lautet: de.ibw.tms.MainTmsSim

Der Main-Entry-Point der SmartLogic lautet: de.ibw.smart.logic.intf.SmartLogic

Zuerst werden die Klassen, die von SmartLogic benutzt werden beschrieben.

## Die SmartLogic

Die Klasse mit den Main-Entry Point heißt SmartLogic

Package de.ibw.smart.logic.intf

### Class SmartLogic

Die SmartLogic-Klasse hat eine Ausgabe-Warteschlange

#### Felder der Smart-Logic-Klasse

|  |
| --- |
| static java.util.concurrent.SynchronousQueue<[SmartServerMessage](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\messages\SmartServerMessage.html)> outputQueue |

Das ist die Ausgangswarteschlange der SL an das TMS. Sie versendet SmartServerMessages.

Weiteres statische Felder dieser SmartLogic-Klasse:

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Erklärung |
| static java.lang.String | **SMART\_LOGIC** | Name der Smart-Logic Komponente für Log-files |

Daneben gibt es ein Timeout-Feld in diese Klasse

|  |
| --- |
| static int [TIMEOUT\_SETTING\_WAITING\_MA\_ACK\_IN\_SECONDS](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\SmartLogic.html#TIMEOUT_SETTING_WAITING_MA_ACK_IN_SECONDS) |

Er gibt an wie lange auf ein Acknowledge vom RBC gewartet wird.

Wenn eine neue MA an das RBC gesendet wird. Wird berücksichtigt, dass die MA nicht bestätigt wird oder bei der Übertragung verloren gegangen sein kann.

#### Methoden der Smart-Logic-Klasse

|  |  |  |
| --- | --- | --- |
| Datentyp | Method | Description |
| static [RbcModul](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\RbcModul.html) | [getRbcClient](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\SmartLogic.html#getRbcClient())() | RBC Modul als Client an den Kommunikationsserver im RBC |
| static [SmartServer](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\SmartServer.html) | [getSmartServForTms](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\SmartLogic.html#getSmartServForTms())() | Server für Anfragen des TMS in der smartLogic |
| static void | [main](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\SmartLogic.html#main(java.lang.String%5B%5D))​(java.lang.String[] args) | Main-Entry-Point der smartLogic |
| static [Balise](file:///C:\Workspace\Javadoc\SL\de\ibw\feed\Balise.html) | [pickRandomBalise](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\SmartLogic.html#pickRandomBalise(int))​(int index) | Eine Funktion zum Testen der Position-Reports.  Der index entspricht der Zug-ID, wird aber nicht unbedingt benötigt. Es wird eine Konsolenausgabe auf der Standardkonsole unter „index“ ausgegeben. |
| static void | [sendDummyPositionReport](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\SmartLogic.html#sendDummyPositionReport())() | Test Utility sendet als RBC einen PositionReport |
| static void | [startSmartLogic](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\SmartLogic.html#startSmartLogic())() | Starten der SL |

#### generateRandomDummyPositionReport

public static void generateRandomDummyPositionReport​(int iCount)

Generiert und schickt „iCount“ viele Dummy-Position-Reports an die SmartLogic

Parameters:

iCount - Anzahl der Position Reports

#### createSmartLogic

public static void createSmartLogic​(boolean bWithUi,

boolean bFakeRbcReceiver,

boolean bSendRbcRequest,

boolean bStartTms)

Intialisiert die Komponenten der SmartLogic

Parameters:

bWithUi - - boolean gibt an ob das TMS mit Oberfäche gestartet wird

bFakeRbcReceiver - - boolean gibt an ob auf der RBC Seite ein Empfänger simuliert werden soll

bSendRbcRequest - - boolean gibt an ob RBC eine Dummy Anfrage an das TMS stellen soll

bStartTms - - boolean gibt an ob das TMS mitgestartet werden soll

#### generatePositionInfo

public static ebd.rbc\_tms.util.PositionInfo generatePositionInfo

Test Utility generiert einen Position-Report

#### startTmsSide

public static void startTmsSide​(boolean bWithUi,

boolean bFakeRbcReceiver,

RbcReceiver remoteRbcReceiver)

Startet das TMS

Parameters:

bWithUi - - boolean ob die Oberfläche gestartet werden soll

bFakeRbcReceiver - - boolean ob die RBC Seite als Empfänger simuliert werden soll

remoteRbcReceiver - - Der RBC Receiver der als Simulation für echte Empfänger dient

Package de.ibw.smart.logic

### Class EventBusManager

Diese Komponente verwaltet Logging GUI Server und das Logging der SL und des TMS

#### Methoden des Event-Bus-Managers

|  |  |  |
| --- | --- | --- |
| Datentyp | Method | Description |
| void | [log](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\EventBusManager.html#log(java.lang.String,java.lang.String))​(java.lang.String sMsg, java.lang.String sModuleId) | Loggt eine Nachricht untern Nennung des angegebenen Untermoduls  sMsg - String - Nachricht, die geloggt werden soll.  sModuleId - - String - Name des Untermoduls, bzw. Komponente. |
| static void | [main](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\EventBusManager.html#main(java.lang.String%5B%5D))​(java.lang.String[] args) | Test Main Methode  args – String[] - unbenutzt |
| static [EventBusManager](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\EventBusManager.html) | [registerOrGetBus](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\EventBusManager.html#registerOrGetBus(int,boolean))​(int id, boolean isTMS) | Factory Methode, die einen EventBusManager generiert, falls das Modul noch keinen Manager besitzt.  Parameters:  id - int - ModulId  isTMS - boolean - ist das Modul ein TMS  Returns:  EventBusManager - gibt den Manager für das Modul der id wider  Throws:  java.io.IOException - - falls es Probleme zum Logger gibt, kann eine Exception erfolgen. |

|  |  |  |
| --- | --- | --- |
| Datentyp | Method | Description |
| static void | [startLogGuiServer](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\EventBusManager.html#startLogGuiServer(boolean))​(boolean bIsTMS) | Startet den GUI-Server unter dem der GUI-Client die Nachrichten abrufen kann.  Parameters:  bIsTMS - boolean - ist der Server innerhalb des TMS  Throws:  java.lang.Exception - - wirft eine Exception wenn der GUIServer nicht generiert werden kann |

Package de.ibw.smart.logic.datatypes

### Enum BlockedArea.BLOCK\_Q\_SCALE

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<BlockedArea.BLOCK\_Q\_SCALE>, java.lang.constant.Constable

Enclosing class:

BlockedArea

public static enum BlockedArea.BLOCK\_Q\_SCALE

extends java.lang.Enum<BlockedArea.BLOCK\_Q\_SCALE>

Dieser Datentyp ist die Einheit der zugeordneten Distanzen. 10CM, 1M und 10M kann verwendet werden. Die Einheiten werden verwendet um blockierte Abschnitte der Blocked-Area einkreisen zu können.

Package de.ibw.smart.logic.datatypes

### Class BlockedArea

public class BlockedArea

extends java.lang.Object

Diese Klasse stellt blockierte Gleisabschnitte dar, aber auch blockierte Knoten, wie zum Beispiel Weichen.

#### Konstruktor von Blocked-Areas

|  |  |
| --- | --- |
| Constructor | Description |
| [BlockedArea](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\datatypes\BlockedArea.html#%3Cinit%3E(de.ibw.tms.ma.physical.TrackElement))​([TrackElement](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\ma\physical\TrackElement.html) Element) | Dieser Konstruktor instanziiert eine Sperrzone auf ein einzelnes Element wie eine Weiche |
| [BlockedArea](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\datatypes\BlockedArea.html#%3Cinit%3E(de.ibw.tms.plan_pro.adapter.topology.TopologyGraph.Edge,de.ibw.smart.logic.datatypes.BlockedArea.BLOCK_Q_SCALE,int,de.ibw.smart.logic.datatypes.BlockedArea.BLOCK_Q_SCALE,int))​([TopologyGraph.Edge](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\plan_pro\adapter\topology\TopologyGraph.Edge.html) E, [BlockedArea.BLOCK\_Q\_SCALE](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\datatypes\BlockedArea.BLOCK_Q_SCALE.html) Start\_Ma, int d\_A\_to\_Block\_Start, [BlockedArea.BLOCK\_Q\_SCALE](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\datatypes\BlockedArea.BLOCK_Q_SCALE.html) End\_Ma, int d\_A\_to\_Block\_End) | Dieser Konstruktor instanziiert eine Sperrzone auf einer Kante E von einem Startpunkt bis zu einem Endpunkt.  Parameters:  E – Kante die Abschnittsweise blockiert wird.  Start\_Ma - - Q\_Scale Einheit der Distanz von A bis zum Beginn der Sperrzone  d\_A\_to\_Block\_Start - - Entfernung bis zum Beginn der Sperrzone von Knoten A  End\_Ma - - Q\_Scale End Einheit der Distanz von A bis zum Ende der Sperzone  d\_A\_to\_Block\_End - - Distanz von A bis zum Ende der Sperrzone |

#### Methoden von der Blocked-Area

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Description |
| boolean | compareIfIntersection(BlockedArea OtherArea | Vergleicht diese Sperrzone mit einer anderen Sperrzone ob es eine Überschneidung gibt.  OtherArea – BlockedArea ist die Sperrzone mit der verglichen wird  Returns true, es gibt eine Überschneidung  False,  es gibt keine Überschneidung |

Package de.ibw.smart.logic.datatypes

### Class QueueUuidMapper

public class QueueUuidMapper

extends java.lang.Object

Diese Klasse verwaltet Warteschlangen für eine Nachricht-UUID (Kommunikationsverlauf). Die UUID wird für die Kommunikation zum RBC verwendet. Man kann dadurch registrieren ob ein RBC schon eine Nachricht empfangen hat, weil dann ein Acknowledge vom RBC gesendet wird.

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Description |
| void | [createQueue](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\datatypes\QueueUuidMapper.html#createQueue(java.util.UUID))​(java.util.UUID uuid) | Diese Methode fügt eine neue Warteschlange für eine UUID zur Verwaltung ein. |
| void | [offer](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\datatypes\QueueUuidMapper.html#offer(java.util.UUID,java.lang.Boolean))​(java.util.UUID uuid, java.lang.Boolean b) | Diese Methode schreibt den Eingang eines Acknowledge in die Warteschlange der gegebenen UUID als Boolean.  Parameters:  uuid - - Der Kommunikationsverlauf  b - - Der Erfolg eines Acknowledge  Throws:  java.lang.InterruptedException - - wird geworfen wenn beim offer ein Interrupt erfolgt |

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| java.lang.Boolean | [poll](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\datatypes\QueueUuidMapper.html#poll(java.util.UUID))​(java.util.UUID uuid) | Diese Methode wartet bis zu einem Timeout auf ein Acknowledge und gibt Erfolg oder ein Fehlen einer Antwort als Boolean zurück.  Parameters:  uuid - - Id der verfolgten Kommunikation  Returns:  Boolean - ist Acknowledge eingegangen  Throws:  java.lang.InterruptedException - - wird geworfen, wenn beim poll der Warteschlange es Interrupt gab |

Package de.ibw.smart.logic.intf

### Class PriorityMessage

All Implemented Interfaces:

java.lang.Comparable<PriorityMessage>

public class PriorityMessage

extends java.lang.Object

implements java.lang.Comparable<PriorityMessage>

Eine Nachricht, die Priority einstellbarer Art hat.

#### Der Konstruktor von Priority-Messages

|  |  |
| --- | --- |
| [PriorityMessage](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\PriorityMessage.html#%3Cinit%3E(ebd.rbc_tms.Message,java.lang.Long))​(ebd.rbc\_tms.Message M, java.lang.Long iPriority) | Konstruktor instanziiert Priority Nachricht mit Inhalt einer Nachricht an das oder aus dem RBC |

#### Methoden von Priority-Messages

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| int | [compareTo](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\PriorityMessage.html#compareTo(de.ibw.smart.logic.intf.PriorityMessage))​([PriorityMessage](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\PriorityMessage.html) priorityMessage) | Priority Vergleich |
| java.lang.Long | [getiPriority](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\PriorityMessage.html#getiPriority())() | Gibt die Wichtigkeit der Nachricht als Long zurück. Je kleiner der Long-Wert desto höher ist die Priority. |
| ebd.rbc\_tms.Message | [getMsg](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\PriorityMessage.html#getMsg())() | Holt die RBC-TMS-Nachricht aus dieser Klasse |
| void | [setiPriority](file:///C:\Workspace\Javadoc\SL\de\ibw\smart\logic\intf\PriorityMessage.html#setiPriority(java.lang.Long))​(java.lang.Long iPriority) | Setzt die Priority als Long |

Package de.ibw.smart.logic.intf

### Class RbcModul

All Implemented Interfaces:

java.lang.Runnable

Direct Known Subclasses:

SmartClient, SmartServer

public class RbcModul

extends java.lang.Thread

Ein RBC Modul sendet mithilfe des Netty-Frameworks Nachrichten an das RBC. Es kann aber auch als Client benutzt werden um Nachrichten aus dem RBC erhalten. Es wird deshalb zweimal instanziiert. Ein RBC-Modul kann entweder als Client oder Server gestartet werden. Es kann aber beliebig viel Instanzen erzeugt werden.

#### Kommunikation RBC-SL

Die Smart-Logic besitzt zwei Module die mit dem RBC reden. Es wird ein RBC-Modul als Server instanziiert, der Nachrichten vom RBC erhält.

Es wird ein weiteres Modul als Client instanziiert, das Nachrichten zum RBC schickt.

Ein Modul kann entweder senden oder empfangen, jedoch nicht beides gleichzeitig. Das war der Grund, warum die SL zwei RBC-Module für die Komunikation zum RBC benötigt.

#### Kommunikation SL-TMS

Der SmartServer ist biderektional. Es gibt einen beständigen Kanal vom TMS zur SL. Die Serverseite befindet sich in der SL. Dieser Server heißt Smart-Server und basiert auf ein erweitertes RBC-Modul.

Das TMS besitzt einen Smart-Client. Ist der Kanal offen, können sowohl das TMS als auch die SL Nachrichten senden und empfangen.

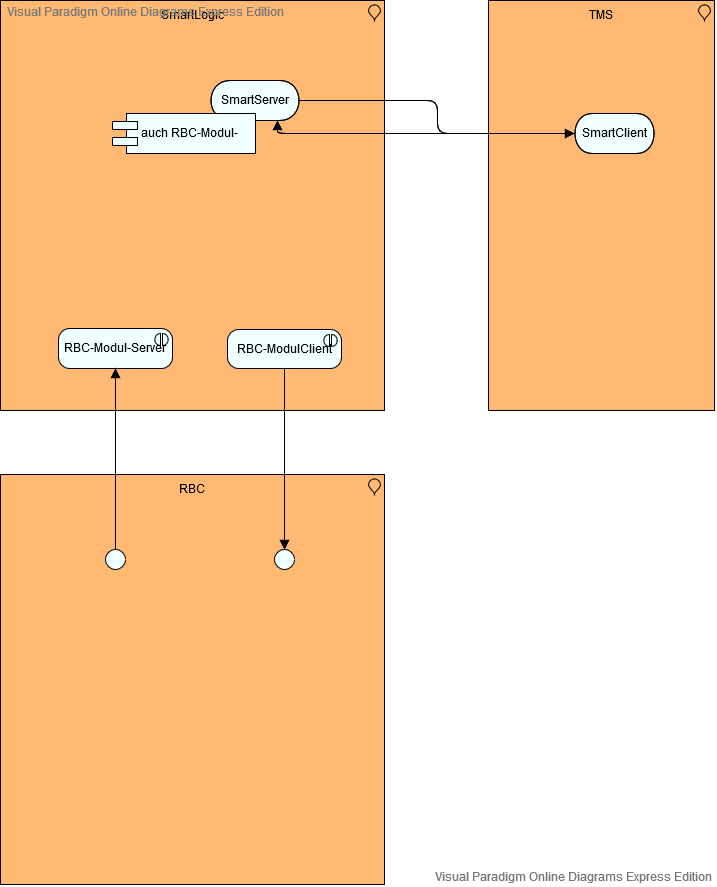


Abbildung 2 Kommikation der beteiligten Module

#### Felder im RbcModul

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Beschreibung |
| int | **iPort** | Port als Integer |
| java.lang.Boolean | **isClient** | Gibt an ob das instanziierte Modul horcht oder sendet – true bedeutet es werden Nachrichten an das RBC gesendet  -false es werden Nachrichten vom RBC empfangen. |
| static java.lang.String | **RBC\_MODUL** | Die Smart Logic verwendet einen Namen dieses Moduls, um Nachrichten loggen zu können, die an das RBC gehen. RBC\_MODUL definiert den Namen im Logging. |
| java.lang.String | **sHost** | Name/IP des Zielservers. |
| static java.lang.String | **TMS\_PROXY** | Die Smart Logic hat ein Modul, dass Nachrichten aus dem RBC bekommt. TMS\_PROXY definiert den Namen für das Logging. |

#### Konstruktor im RbcModul

**public RbcModul**​(boolean bIsClient,

java.lang.String sHost,

int iPort)

Kann als Server oder als Client des RBCs instanziiert werden. Der Server dient ausschließlich zum Empfang von Nachrichten des RBCs. Der Client dient ausschließlich zum Senden von Nachrichten an das RBC.

Parameters:

bIsClient - - boolean der definiert ob als Client instanzirt.

sHost - - String des Hostnamens/IP

iPort - - Port der Kommunikation

**public RbcModul**​(ebd.rbc\_tms.Message M,

java.lang.String sHost,

int iPort)

Senden einzelner Nachricht nur für Testzwecke

Parameters:

M - - Nachricht zu senden

sHost - - Host Bezeichnung gibt es in Netty auch als Angabe im Server-Modus, das ist aber Netty-Spezifisch

iPort - - Port auf dem gehorcht oder gesendet wird

#### Die Methoden des RbcModul

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| void | [run](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\RbcModul.html#run())() | Es gibt zwei Instancen des RBC-Moduls.  Der SERVER startet einen stellvertretenen TMS Server innerhalb der SL für das RBC. Dadurch stellt die SL einen Server für das RBC bereit. Der CLIENT sendet das RBC Nachrichten aus der Queue an das RBC. Jedoch wird immer ein Kommunikationskanal pro Nachricht auf und abgebaut. |
| void | [sendMessage](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\RbcModul.html#sendMessage(de.ibw.smart.logic.intf.PriorityMessage))​([PriorityMessage](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\PriorityMessage.html) priorityMessage) | Sendet Nachricht an das RBC  Parameters:  priorityMessage - - Priority-Message, weil in Warteschlange Priority geachtet wird.  Die niedrigste Priority wird al nächstes gesendet. |
| void | [setTmsHandler](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\RbcModul.html#setTmsHandler(de.ibw.smart.logic.intf.TmsIntf))​([TmsIntf](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\TmsIntf.html) tmsHandler) | Gibt an wer für dieses Modul als TMS handelt. Bisher hat die SmartLogic einen Proxy, der so tut als sei er selbst das TMS und alle Nachrichten zwischenliest.  Parameters:  tmsHandler - - Parameter, der bestimmt an welche Komponente gelesenen RBC-Daten weitergegeben werden. |
| void | [start](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\RbcModul.html#start())() | Startet dieses RBC Modul |

**public static ebd.rbc\_tms.message.Message\_00** createResponseMessage​(int iErrorCode,

java.lang.String rbc\_id,

java.util.UUID uuid,

java.lang.String tms\_id,

ebd.rbc\_tms.util.TrainInfo TI)

generiert eine Antwort, auf vorher erhaltene Nachrichten.

Parameters:

iErrorCode - - gibt an ob mit der vorherigen Nachricht ein Fehler verbunden ist

rbc\_id - - ID RBC

uuid - - ID der Kommunikation

tms\_id - - ID des TMS

TI - - Informationen zum Zug

Returns:

Message\_00 - eine Antwortnachricht

**public static void**startTcpClient​(java.lang.String sHost,

int iPort,

io.netty.channel.SimpleChannelInboundHandler customHandlerTcpClient,

ebd.rbc\_tms.Message Msg)

throws java.lang.InterruptedException

Diese Methode startet das Modul als Tcp-Client über Netty.

Parameters:

sHost - - IP des Zielhosts, mit dem sich verbunden werden soll

iPort - - Ziel-Port

customHandlerTcpClient - - Kann null sein, um eine Standard-Implementierung zu benutzten. Bisher wird der Tcp-Client nur mit null als diesem Parameter genutzt.

Msg - - Nachricht die gesendet werden soll

Throws:

java.lang.InterruptedException - - Kommunikationsfehler mit Netty-Client

**public void**startServer​(java.lang.String sHost,

int iPort,

io.netty.channel.ChannelInboundHandlerAdapter ServerHandler)

throws java.lang.InterruptedException

Diese Methode des RBC-Modules wird nicht für Kommunikation mit dem RBC benutzt.

Diese Methode startet in der SmartLogic einen Server, der biderektional mit einem TMS kommuniziert. Dieser Channel wird aufrechterhalten.

Parameters:

sHost - - localhost - nicht relevant bei server, aber Netty hat diese Option

iPort - - port dieses Servers

ServerHandler - - Definiert was der Server mit ein- und ausgehenden Daten unternimmt und weiterleitet

Throws:

java.lang.InterruptedException - - Fehler im Kommunikationskanal

**public void** startTmsServer​(java.lang.String sHost,

int iPort)

throws java.lang.InterruptedException

SL stellt Verbindung für das RBC als Server bereit

Parameters:

sHost - - Netty hat eine Host Definition, auch für Server – STD: „localhost“

iPort - - port auf dem der Server arbeitet oder horcht

Throws:

java.lang.InterruptedException - - Kommunikation hatte einen Fehler verursacht

Es folgen Unterkomponenten des Rbc-Moduls.

Package de.ibw.smart.logic.intf

#### Class RbcModul.SL\_To\_RBC\_ClientThread

Diese Klasse instanziiert einen Thread und erbt deshalb von java.lang.Thread

All Implemented Interfaces:

java.lang.Runnable

Enclosing class:

RbcModul

public static class RbcModul.SL\_To\_RBC\_ClientThread

extends java.lang.Thread

Diese Klasse stellt einen Thread dar, der eine Nachricht an das RBC sendet. Nachdem Senden wird der Thread wieder entfernt. Für die nächste Nachricht wird wieder ein neuer Thread dieser Klasse erstellt.

##### Der Konstruktor des Rbc-Client-Threads:

public SL\_To\_RBC\_ClientThread​(java.lang.String sHost,

int iPort,

ebd.rbc\_tms.Message M)

Instanziierrt einen Thread mit einer Nachricht an das RBC

Parameters:

sHost - - Ziel RBC als IP

iPort - - Port des Servers des Ziels

M - - Nachricht, die gesendet werden soll

##### Die Run-Methode des RBC-Client-Threads:

public void run()

Sendet Nachricht an das RBC

Package de.ibw.smart.logic.intf

#### Class RbcModul.TcpClientHandler

Diese Klasse ist ein Channel-Inbound-Handler von Netty und erbt deshalb von:

io.netty.channel.SimpleChannelInboundHandler

All Implemented Interfaces:io.netty.channel.ChannelHandler, io.netty.channel.ChannelInboundHandler

Enclosing class:

RbcModul

public static class RbcModul.TcpClientHandler

extends io.netty.channel.SimpleChannelInboundHandler

##### Der Konstruktor des Tcp-Client-Handlers:

public TcpClientHandler​(ebd.rbc\_tms.Message Msg)

Netty Client Handler zum Schicken von Nachrichten ans RBC

Parameters:

Msg - - Nachricht die geschickt wird

##### Die Methoden des Tcp-Client-Handlers

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| void | [channelActive](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\RbcModul.TcpClientHandler.html#channelActive(io.netty.channel.ChannelHandlerContext))​(io.netty.channel.ChannelHandlerContext channelHandlerContext) | Kanal in Netty durch den das RBC die Nachricht bekommt |
| protected void | [channelRead0](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\RbcModul.TcpClientHandler.html#channelRead0(io.netty.channel.ChannelHandlerContext,java.lang.Object))​(io.netty.channel.ChannelHandlerContext channelHandlerContext, java.lang.Object sJsonMsg) | Eigentlich unbenutzt, weil zum Empfang, eine andere Client Server Kommunikation verwendet wird. |
| void | [exceptionCaught](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\RbcModul.TcpClientHandler.html#exceptionCaught(io.netty.channel.ChannelHandlerContext,java.lang.Throwable))​(io.netty.channel.ChannelHandlerContext channelHandlerContext, java.lang.Throwable cause) | Handler wenn Fehler in Kommunikation zum RBC |

Package de.ibw.smart.logic.intf

#### Class RbcModul.TmsServerHandler

Diese Klasse wird von einem Server benutzt, der Nachrichten vom RBC erhält.

Um Netty-Funktionen zu nutzen, erbt die Klasse von:

io.netty.channel.ChannelInboundHandlerAdapter

All Implemented Interfaces:

io.netty.channel.ChannelHandler, io.netty.channel.ChannelInboundHandler

Enclosing class:

RbcModul

public class RbcModul.TmsServerHandler

extends io.netty.channel.ChannelInboundHandlerAdapter

Netty Handler eines Servers der SL zur Kommunikation mit RBC. Dieser Handler sendet keine Nachricht. Es werden nur Nachrichten vom RBC entgegengenommen. Der Name Tms-Server-Handler kommt daher, dass dem RBC vorgetäuscht wird, dass es mit einem TMS kommuniziert. In Wirklichkeit wird der TMS-Server-Handler in der Smart-Logic betrieben und die Smart-Logic sendet nach einer Verarbeitung die Nachricht weiter an das eigentliche TMS. Es wird durch einen Kanal immer nur eine Nachricht gesendet. Danach muss ein neuer Kanal mit der SL benutzt werden.

Felder des Tms-Server-Handlers

public static final java.lang.String TMS\_HANDLER

Diese Konstante bezeichnet dieses Untermodul in der Smart-Logic mit einem Namen, der im Log, verwendet wird.

##### Die Methoden des Tcp-Server-Handlers

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| void | [channelRead](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\RbcModul.TmsServerHandler.html#channelRead(io.netty.channel.ChannelHandlerContext,java.lang.Object))​(io.netty.channel.ChannelHandlerContext ctx, java.lang.Object msg) | Definiert was passiert, wenn Netty eine Nachricht vom RBC bekommt |
| void | [channelReadComplete](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\RbcModul.TmsServerHandler.html#channelReadComplete(io.netty.channel.ChannelHandlerContext))​(io.netty.channel.ChannelHandlerContext ctx) | Channel wird wieder geschlossen, wenn eine Nachricht erhalten wurde. |
| void | [exceptionCaught](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\RbcModul.TmsServerHandler.html#exceptionCaught(io.netty.channel.ChannelHandlerContext,java.lang.Throwable))​(io.netty.channel.ChannelHandlerContext ctx, java.lang.Throwable cause) | Fehler Handler, wenn der Netty Server gegenüber dem RBC Kommunikations-Fehler registriert |

Package de.ibw.smart.logic.intf

### Class SmartLogicTmsProxy

All Implemented Interfaces:

TmsIntf

Direct Known Subclasses:

SmartServer4TmsImpl

public class SmartLogicTmsProxy

extends java.lang.Object

implements TmsIntf

Die Smart Logic simuliert das TMS als Proxy. Diese Komponente stellt den Proxy auf ein TMS innerhalb der SL dar. Wenn das RBC eine Nachricht an die SL schickt, leitet das die SL an dessen Smart-Server weiter. Der Smart-Server übernimmt die Kommunikation mit dem TMS.

#### Felder im SmartLogic-Tms-Proxy

|  |  |  |
| --- | --- | --- |
| static java.lang.String | **TMS\_PROXY** | Die Komponentenbezeichnung dieses Moduls in Log-Ausgaben |

#### Konstruktor des SmartLogic-Tms-Proxy

public SmartLogicTmsProxy​(SmartServer smartServer4Tms)

Konstruktor der den Proxy instanziiert.

Parameters:

smartServer4Tms - - das Server Modul in der SmartLogic die den Proxy benutzt

#### Methoden des SmartLogic-Tms-Proxys

public void setiPriority​(long lPriority)

Setzen der Priority von Nachrichten des Proxies an das TMS

Parameters:

lPriority - long – Priority die niedrigste Priority wird zuerst gesendet.

public void setTmsServer​(RbcModul TmsServer)

Wird im Proxy nicht aufgerufen, weil er zur Laufzeit nicht die Komponente (TMS) wechselt, an die Nachrichten weitergeleitet werden. Der TMSServer erhält Nachrichten vom RBC über diesen Proxy

Parameters:

TmsServer - RBCModul -not called

public ebd.rbc\_tms.Message handleMaRequest​(ebd.rbc\_tms.Message<ebd.rbc\_tms.Payload> msgFromRbc)

verwaltet was der Proxy unternimmt, wenn eine Ma-Request vom RBC eintrifft. Bisher wird die Nachricht an das TMs weitergeleitet.

Parameters:

msgFromRbc - Message - Nachricht des RBC

Returns:

Message - Standard Acknowledgment wird an des RBC als Antwort gesendet

public ebd.rbc\_tms.Message handleRegister​(ebd.rbc\_tms.Message<ebd.rbc\_tms.Payload> msgFromRbc)

verwaltet was der Proxy unternimmt, wenn eine Registrierung vom RBC eintrifft. Bisher wird die Nachricht an das TMS weitergeleitet.

Parameters:

msgFromRbc - Message - Nachricht des RBC

Returns:

Message - Standard Acknowledgment wird an des RBC als Antwort gesendet

public ebd.rbc\_tms.Message handleLogin​(ebd.rbc\_tms.Message<ebd.rbc\_tms.Payload> msgFromRbc)

verwaltet was der Proxy unternimmt, wenn eine Anmeldung vom RBC eintrifft. Bisher wird die Nachricht an das TMS weitergeleitet.

Parameters:

msgFromRbc - Message - Nachricht des RBC

Returns:

Message - Standard Acknowledgment wird an des RBC als Antwort gesendet

public void handleNoError​(ebd.rbc\_tms.Message<ebd.rbc\_tms.Payload> msgFromRbc)

Verwaltet was der Proxy unternimmt, wenn eine Nachricht vom RBC eintrifft, die anzeigt, dass das RBC eine Anfrage ohne Fehler entgegengenommen hat. Es wird dabei die SmartLogic das Ack verwalten und dann an das TMS weiterleiten.

Parameters:

msgFromRbc - Message - Nachricht des RBC

public ebd.rbc\_tms.Message handlePositionReport​(ebd.rbc\_tms.Message msgFromRbc)

verwaltet was der Proxy unternimmt, wenn ein PositionReport vom RBC eintrifft. Der Position Report wird vom SmartLogic Server mituntersucht

Parameters:

msgFromRbc - Message - Nachricht des RBC

Returns:

Message - Acknowledgment wird an des RBC als Antwort gesendet - kann auch einen Fehler senden

Package de.ibw.smart.logic.intf

### Class SmartServer

All Implemented Interfaces:

java.lang.Runnable

public class SmartServer

extends [RbcModul](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\RbcModul.html)

Der Smart Server ist ein Server für die Kommunikation der SL zum TMS. Die SL hat in dieser Kommunikation den Serverpart. Die Verbindung wird aufrecht gehalten.

#### Konstruktor des Smart-Servers

public SmartServer​(java.lang.String sHost,

int iPort)

Konstruktor zum Starten des Smart Servers zur Verwaltung von Anfragen an das SL vom TMS.

Parameters

sHost - - null, da der Server nicht den Host vorgibt

iPort - - int, Port auf dem dieser Server kommuniziert

#### Methoden des Smart-Servers

public static void setTmsOutChannel​(io.netty.channel.Channel C)

Setzt den Channel über den die Smart-Logic mit dem TMS kommunizieren.

public void run()

Startet einen Netty Server mit dem SmartHandler dieses Servers

Specified by:

run in interface java.lang.Runnable

Overrides:

run in class RbcModul

public void start()

Startet diesen Server als Thread

Overrides:

start in class RbcModul

Package [de.ibw.smart.logic.intf.impl](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\package-summary.html)

### Class SmartServer4TmsImpl

All Implemented Interfaces:

[SmartServerFromTmsIntf](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\SmartServerFromTmsIntf.html), [TmsIntf](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\TmsIntf.html)

public class SmartServer4TmsImpl

extends [SmartLogicTmsProxy](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\SmartLogicTmsProxy.html)

implements [SmartServerFromTmsIntf](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\SmartServerFromTmsIntf.html), [TmsIntf](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\TmsIntf.html)

Diese Klasse verwaltet die Serverroutine des SmartLogic-Servers.

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Erklärung |
| static [QueueUuidMapper](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\datatypes\QueueUuidMapper.html) | [ackQueues](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#ackQueues) | Dieser Mapper verwaltet Warteschlangen für Acknowledges vom RBC |
| static java.lang.String | [ELEMENT\_RESERVATION\_ERROR](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#ELEMENT_RESERVATION_ERROR) | Einige Routen-Elemente für das Freigaben einer MA durch die SmartLogic sind belegt |
| static [SmartServer4TmsImpl](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html) | [instance](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#instance) | Instanz dieses Servers |
| static java.lang.String | [NO\_ACK](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#NO_ACK) | Die weitergegebene MA wurde vom RBC nicht bestätigt |
| static java.lang.String | [NO\_DANGERZONE\_RETRIEVAL\_ERROR](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#NO_DANGERZONE_RETRIEVAL_ERROR) | Gefahrzonen konnten nicht abgerufen werden |
| static java.lang.String | [NOT\_ALL\_ELEMENTS\_GIVEN\_FOR\_RESERVATION\_ERROR](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#NOT_ALL_ELEMENTS_GIVEN_FOR_RESERVATION_ERROR) | Die Liste der Routen Elemente ist nicht durchgängig definiert. |
| static java.lang.String | [ROUTE\_COMPONENTS\_IDENTIFY](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#ROUTE_COMPONENTS_IDENTIFY) | Modulbezeichnung der Checkroutine welche Routen-Elemente vorliegen für das Logging |
| static java.lang.String | [SL\_SELF\_CHECK\_ERROR](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#SL_SELF_CHECK_ERROR) | Selbstcheck der Smartlogic bereitet Fehler |
| static java.lang.String | [SMART\_SERVER\_MA\_MODUL](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#SMART_SERVER_MA_MODUL) | Modulbezeichnung von der MA-Verwaltung dieses Servers für das Logging |
| static java.lang.String | [SSP\_CHECK\_ERROR](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#SSP_CHECK_ERROR) | Das Geschwindigkeitsprofil ist nicht zulässig |
| static java.lang.String | [STATUS\_OF\_ELEMENT\_IS\_WRONG\_ERROR](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#STATUS_OF_ELEMENT_IS_WRONG_ERROR) | Ein Element hat einen Status, sodass die Route nicht von der SL freigegeben werden kann |
| static java.lang.String | [TMS\_NOT\_IDENTIFIED\_ERROR](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#TMS_NOT_IDENTIFIED_ERROR) | TMS konnte nicht von der SL identifiziert werden |

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Erklärung |
| static java.lang.String | [TRAIN\_ID\_NOT\_VERIFIED\_ERROR](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#TRAIN_ID_NOT_VERIFIED_ERROR) | Zug-Id konnte nicht verifiziert werden |
| static java.lang.String | [TRAIN\_NOT\_FULFIL\_ROUTE\_CRITERIA\_ERROR](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\SmartServer4TmsImpl.html#TRAIN_NOT_FULFIL_ROUTE_CRITERIA_ERROR) | Der Zug in seinem Status kann die Anforderung nicht ausführen |

#### Konstruktor der SmartServer4TmsImpl

public SmartServer4TmsImpl​([RbcModul](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\RbcModul.html) rbcCl,

[SmartServer](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\SmartServer.html) smartServer)

Diese Implementierung des SmartServers hat ein Modul zum Empfangen von Nachrichten des RBC als RBC-Modul. Diese Implementierung hat eine Serverkomponente, die mit dem TMS kommuniziert.

Parameters:

rbcCl - - RBC zum Senden von Nachrichten an das RBC

smartServer - - Server der Anfragen des TMS verwaltet

#### Methode der SmartServer4TmsImpl

public void checkMovementAuthority​([MaRequestWrapper](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\tms\ma\MaRequestWrapper.html) MaRequest,

ebd.rbc\_tms.util.MA Ma,

java.util.UUID uuid,

java.lang.String tms\_id,

java.lang.String rbc\_id,

java.lang.Long lPriority)

Diese Methode checkt die Eingegangen MA vom TMS

Parameters:

MaRequest - - ein Request aus TMS

Ma - - eine MA zum RBC

uuid - - Kommunikations-Id mit RBC

tms\_id - - Id des TMS

rbc\_id - - Id des angesprochenen RBCs

lPriority - - Priority als long, die niedrigste Priorität wird zuerst gesendet.

Package de.ibw.smart.logic.intf.impl.threads

### Class TmsOuputWorker<T>

All Implemented Interfaces:

java.lang.Runnable

public class TmsOuputWorker<T>

extends java.lang.Thread

Diese Klasse verwaltet die Senderoutine sowohl aus TMS zu SL als auch von SL zum TMS. Der Nachrichtentyp wird durch den Templateparameter T bestimmt.

#### Felder im Tms-Output-Worker

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| static java.lang.String | [SENDING\_MODUL](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\threads\TmsOuputWorker.html#SENDING_MODUL) | Bezeichnung dieses Moduls für das Logging |
| static [TmsOuputWorker](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\threads\TmsOuputWorker.html)<[SmartServerMessage](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\SmartServerMessage.html)> | [SmartToTmsWorker](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\threads\TmsOuputWorker.html#SmartToTmsWorker) | Senderoutine von SL zum TMS |
| static [TmsOuputWorker](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\threads\TmsOuputWorker.html)<java.lang.String> | [TmsToSmartWorker](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\threads\TmsOuputWorker.html#TmsToSmartWorker) | Senderoutine vom TMS zur SL |

#### Konstruktor eines Tms-Output-Worker

public TmsOuputWorker​(java.util.concurrent.SynchronousQueue<[T](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\impl\threads\TmsOuputWorker.html)> queue,

io.netty.channel.ChannelHandlerContext ctx)

Konstruktor zum Instanziieren einer Senderoutine

Parameters:

queue - - Warteschlange für Nachrichtenausgang

ctx - - Kontext der Nachrichten einstellt

#### Methode eines Tms-Output-Worker

public void run()

Start der Senderoutine

Specified by:

run in interface java.lang.Runnable

Overrides:

run in class java.lang.Thread

Package de.ibw.smart.logic.intf.messages

### Class MaRequestReturnPayload

public class MaRequestReturnPayload

extends java.lang.Object

implements [ITypable](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\ITypable.html)

Diese Klasse stellt eine Antwort des SL an das TMS für eine Ma-Anforderung dar.

#### Feld im Ma-Request-Return-Payload

|  |  |  |
| --- | --- | --- |
| static java.lang.String | [RETURN\_TYPE](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\MaRequestReturnPayload.html#RETURN_TYPE) | Definiert Nachrichtentyp, dadurch kann die Art der Nachricht vom TMS erkannt werden |

#### Methoden des Ma-Request-Return-Payload

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| java.util.ArrayList<java.lang.String> | [getFailureCodes](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\MaRequestReturnPayload.html#getFailureCodes())() | Gibt Liste von Fehlercodes zurück, die bei der MA-Bearbeitung der Smart-Logic entstanden sind. |
| java.lang.String | [getType](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\MaRequestReturnPayload.html#getType())() | Typ-Kennung der Art dieser Nachricht. |
| java.util.UUID | [getUuid](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\MaRequestReturnPayload.html#getUuid())() | Id des Kommunikationsverlaufes mit RBC |
| boolean | [isMaSuccessfull](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\MaRequestReturnPayload.html#isMaSuccessfull())() | Die Ma war erfolgreich und wurde im RBC bestätigt |
| boolean | [isUnlockElements](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\MaRequestReturnPayload.html#isUnlockElements())() | Wurden Elemente freigegeben  true – Freigabe  false – keine Freigabe |
| java.lang.String | [parseToJson](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\MaRequestReturnPayload.html#parseToJson())() | Baut eine Json-Zeichenkette aus exposed Felder dieser Klasse |
| void | [setErrorState](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\MaRequestReturnPayload.html#setErrorState(java.util.UUID,boolean,java.lang.String))​(java.util.UUID uuid, boolean unlockElements, java.lang.String sFailureCode) | Diese Method setzt diese Nachricht in Fehlerstatus, weil die Ma-Anforderung nicht erfolgreich war |
| void | [setMaSuccessfull](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\MaRequestReturnPayload.html#setMaSuccessfull(java.util.UUID))​(java.util.UUID uuid) | Diese Antwort als Erfolg setzen: |

Package de.ibw.smart.logic.intf.messages

### Class SmartServerMessage

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<SmartServerMessage>

public class SmartServerMessage

extends java.lang.Object

implements java.lang.Comparable<SmartServerMessage>, java.io.Serializable

Diese Klasse stellt eine Nachricht von SL zum TMS dar oder umgekehrt. Sie ist priorisierbar und implementiert deshalb Comparable.

#### Konstruktor einer Smart-Server-Message

public SmartServerMessage​(java.lang.String M,

java.lang.Long iPriority)

Dieser Konstruktor instanziiert die Nachricht mit Json-String M und einer Priority

Parameters:

M - String – Json das gesendet wird

iPriority – die Nachricht mit der niedrigsten Priority wird gesendet

#### Methoden einer Smart-Server-Message

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| int | [compareTo](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\SmartServerMessage.html#compareTo(de.ibw.smart.logic.intf.messages.SmartServerMessage))​([SmartServerMessage](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\SmartServerMessage.html) priorityMessage) | Vergleicht welche Nachricht mehr Priorität. Die geringste Priorität wird als erstes gesendet. |
| static [ITypable](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\ITypable.html) | [generateFromSlJson](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\SmartServerMessage.html#generateFromSlJson(java.lang.String))​(java.lang.String jsonString) | Diese Methode gibt den Json-String typgerecht als Nachrichtenobjekt wieder.  ITypable ist die allgemeine Nachrichtenform |
| java.lang.Long | [getiPriority](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\SmartServerMessage.html#getiPriority())() | Gibt die Priority dieser Nachricht wieder |
| java.lang.String | [getMsg](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\SmartServerMessage.html#getMsg())() | Holt die Nachricht als Json-Zeichenkette aus diesem Wrapper |
| boolean | [isbIsFromSL](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\SmartServerMessage.html#isbIsFromSL())() | Gibt an ob Nachricht zum TMS gesendet wird oder an die SmartLogic |
| void | [setbIsFromSL](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\SmartServerMessage.html#setbIsFromSL(boolean))​(boolean bIsFromSL) | Setzt die Richtung.  bIsFromSl – parameter  gibt an welche Richtung vorliegt. |
| void | [setiPriority](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\SmartServerMessage.html#setiPriority(java.lang.Long))​(java.lang.Long iPriority) | Setzt die Priorität dieser Nachricht |
| java.lang.String | [toString](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\intf\messages\SmartServerMessage.html#toString())() | als Zeichenkette wird der Json-String widergegeben |

Package de.ibw.smart.logic.safety

### Class SmartSafety

public class SmartSafety

extends java.lang.Object

Dieses Modul übernimmt alle Überprüfungen, die die SmartLogic bei eingehenden Anfragen des TMS unternehmen muss.

#### Felder in der Smart-Safety

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Beschreibung |
| static java.lang.String | [SELF\_CHECK](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\safety\SmartSafety.html#SELF_CHECK) | Untermodul der Selbstuntersuchung der Smart-Logic |
| static java.lang.String | [SMART\_SAFETY](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\safety\SmartSafety.html#SMART_SAFETY) | Modulname der Smart-Safety im Logging |
| static java.lang.String | [TRACK\_SAFETY](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\safety\SmartSafety.html#TRACK_SAFETY) | Untermodul der Routenanalyse der Smart-Safety im Logging |
| static [DefaultRepo](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\util\DefaultRepo.html)<java.lang.Integer,​ebd.rbc\_tms.util.TrainInfo> | [trainInformation](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\safety\SmartSafety.html#trainInformation) | Ein Repository, dass durch eine ZugId, die Zugdaten zum Zug speichert. |
| static java.lang.String | [UNKNOWN\_TRACK\_ELEMENT\_GIVEN](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\smart\logic\safety\SmartSafety.html#UNKNOWN_TRACK_ELEMENT_GIVEN) | Nachricht, wenn ein noch nicht definiertes Routen-Element besteht. |

**public static**[**DefaultRepo**](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\util\DefaultRepo.html)**<java.lang.Integer,​ebd.rbc\_tms.util.PositionInfo>** lastPositionReport

Ein Repository, dass durch eine ZugId die Positionsdaten des Zuges speichert.

**public static**[**DefaultRepo**](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\util\DefaultRepo.html)**<java.lang.Integer,​org.apache.commons.collections.buffer.CircularFifoBuffer>** positionHistory

Ein Repository, dass durch eine ZugId, vergangene Positionsdaten speichert.

**public static**[**DefaultRepo**](file:///C:\Workspace\tms_simulator\misc\docs\SL_HTML_Dokumentatkion\SL\de\ibw\util\DefaultRepo.html)**<java.lang.Integer,​java.lang.Integer>** passedMetersSinceLrbg

Ein Repository, das für eine Zug Id, die Meter ab der letzten Balise speichert

#### Methoden der Smart-Safety

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| boolean | [checkIfTrainStatusRequestIsFresh](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/SmartSafety.html#checkIfTrainStatusRequestIsFresh(de.ibw.tms.ma.MaRequestWrapper))​([MaRequestWrapper](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/ma/MaRequestWrapper.html) maRequest) | Noch nicht implementiert |
| boolean | [checkSSP](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/SmartSafety.html#checkSSP(ebd.rbc_tms.util.MA))​(ebd.rbc\_tms.util.MA ma) | Noch nicht implementiert |
| boolean | [checkTrainForRouteCriteria](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/SmartSafety.html#checkTrainForRouteCriteria())() | Noch nicht implementiert |
| boolean | [dangerZonesAreReceivable](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/SmartSafety.html#dangerZonesAreReceivable())() | Noch nicht implementiert |
| static [SmartSafety](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/SmartSafety.html) | [getSmartSafety](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/SmartSafety.html#getSmartSafety())() | Ein Singleton, dass Sicherheitslogic der SmartLogic wiedergibt. |
| boolean | [slSelfCheck](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/SmartSafety.html#slSelfCheck(de.ibw.tms.ma.RbcMaAdapter))​([RbcMaAdapter](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/ma/RbcMaAdapter.html) maAdapter) | Untersucht ob die SmartLogic richtig funktioniert  Parameters:  maAdapter - RbcMaAdapter - Daten die zum RBC gesendet werden sollen, wenn die Anfrage ok ist.  Returns:  boolean - funktioniert SL |
| boolean | [verifyTms](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/SmartSafety.html#verifyTms(java.lang.String))​(java.lang.String tms\_id) | Noch nicht implementiert |
| boolean | [verifyTrainID](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/SmartSafety.html#verifyTrainID(de.ibw.tms.ma.MaRequestWrapper,ebd.rbc_tms.util.MA))​([MaRequestWrapper](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/ma/MaRequestWrapper.html) maRequest, ebd.rbc\_tms.util.MA ma) | Noch nicht implementiert |

**public static** SmartSafety getSmartSafety()

Ein Singleton, dass Sicherheitslogic der SmartLogic wiedergibt.

Returns:

SmartSafety - Die Sicherheitsanalyse der SmartLogic

**public boolean** checkIfRouteIsNonBlocked​(MaRequestWrapper maRequest,

RbcMaAdapter maAdapter,

java.util.ArrayList<org.apache.commons.lang3.tuple.Pair<Route.TrackElementType,​TrackElement>> requestedTrackElementList)

Methode, die untersucht ob auf der angeforderten MA blockierte Streckenabschnitte bestehen. Es kann zum Beispiel durch einen anderen Zug auf der Strecke, blockierte Abschnitte bestehen.

Parameters:

maRequest - - MaRequestWrapper - Anfragedaten zur MA

maAdapter - RbcMaAdapter - Daten, die zum RBC gesendet werden sollen, wenn die Anfrage ok ist.

requestedTrackElementList - ArrayList - Eine Liste der Routenelemente die nicht blockiert sein sollten. Das untersucht diese Methode.

Returns:

true - hat die Route keine blockierten Elemente oder Abschnitte

false – die Strecke der untersuchten MA hat blockierte Elemente oder Abschnitte

**public boolean** checkIfRouteIsContinuousConnected​(MaRequestWrapper maRequest,

java.util.ArrayList<org.apache.commons.lang3.tuple.Pair<Route.TrackElementType,​TrackElement>> requestedTrackElementList)

Diese Methode untersucht ob alle Routenelemente durchwegs verbunden sind.

Parameters:

maRequest - - MaRequestWrapper - Anfragedatend des TMS

requestedTrackElementList - - ArrayList - Eine Liste der Routenelemente die verbunden sein sollten.

Returns:

boolean - ist Route durchwegs verbunden – true

eine Route ist nicht durchwegs verbunden - false

**public boolean** checkIfRouteElementStatusIsCorrect​(MaRequestWrapper maRequest,

java.util.ArrayList<org.apache.commons.lang3.tuple.Pair<Route.TrackElementType,​TrackElement>> requestedTrackElementList)

Noch nicht implementiert. Untersucht ob alle Status der Streckenelemente in Ordnung sind.

Parameters:

maRequest - - MaRequestWrapper - Anfragedaten zur MA

requestedTrackElementList - - ArrayList - Eine Liste der Routenelemente die Statusuntersuchung erfordern.

Returns:

boolean – alle Status erlauben ein Freigeben der MA – true

Status erlauben kein Ausführen einer MA - false

**public boolean** handlePositionReport​(ebd.rbc\_tms.Message<ebd.rbc\_tms.payload.Payload\_14> PositionReport)

Untersucht ob der PositionReport valide ist

Parameters:

PositionReport - Message - RBC - PositionReport

Returns:

boolean - ist Positionsangabe valide – true

ist die Positionsangabe nicht valide - false

**public boolean** checkIfTrainStatusRequestIsFresh​(MaRequestWrapper maRequest)

Noch nicht implementiert

Parameters:

maRequest -

Returns:

**public void** handleFlankProtection​([MaRequestWrapper](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/ma/MaRequestWrapper.html) maRequest,

java.util.ArrayList<org.apache.commons.lang3.tuple.Pair<[Route.TrackElementType](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/ma/Route.TrackElementType.html),​[TrackElement](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/ma/physical/TrackElement.html)>> requestedTrackElementList)

Noch nicht implementiert

Package [de.ibw.smart.logic.safety.self.tests](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/self/tests/package-summary.html)

### Class SmartSafetyContinousConnectTest

Dieser Test kontrolliert ein Modul in der SmartLogic. Das Modul in der Smart-Safety untersucht, ob alle Routen-Elemente kontinuierlich verbunden sind. Die folgenden Methoden sind JUnit-Testmethoden.

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| void | [checkIfMainNullError](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/self/tests/SmartSafetyContinousConnectTest.html#checkIfMainNullError())() | Kontrolliert, dass die Routenliste nicht null sein darf |
| void | [checkIfRouteContinousShallReturnTrue](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/self/tests/SmartSafetyContinousConnectTest.html#checkIfRouteContinousShallReturnTrue())() | Kontrolliert, ob SmartSafety erkennt, dass die Route in Ordnung ist |
| void | [checkIfRouteIsContinuousWithEmptyTrackList](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/self/tests/SmartSafetyContinousConnectTest.html#checkIfRouteIsContinuousWithEmptyTrackList())() | Kontrolliert, dass die Routenliste nicht leer sein darf |
| void | [checkIfRouteIsContinuousWithToLessItems](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/self/tests/SmartSafetyContinousConnectTest.html#checkIfRouteIsContinuousWithToLessItems())() | Kontrolliert das sie SmartSafty erkennt, dass zu wenig Routenelemente gegeben sind. |
| void | [checkIfSmartSaftyRecogniseInvalidNullValues](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/self/tests/SmartSafetyContinousConnectTest.html#checkIfSmartSaftyRecogniseInvalidNullValues())() | Kontrolliert, ob SmartSafety erkennt, dass ein Routen-Element in der Liste null ist. |
| void | [checkRouteHavingWrongEnd](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/self/tests/SmartSafetyContinousConnectTest.html#checkRouteHavingWrongEnd())() | Kontrolliert, dass die SmartSafety erkennt, dass das End-Element nicht in Ordnung ist |
| void | [checkRouteHavingWrongStart](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/safety/self/tests/SmartSafetyContinousConnectTest.html#checkRouteHavingWrongStart())() | Kontrolliert, dass die SmartSafety erkennt, dass das Start-Element nicht in Ordnung ist |

## Der TMS-Simulator

Die Klasse mit dem Main-Entry-Point des TMS-Simulators heißt MainTmsSim.

### Class MainTmsSim

Diese Klasse ist der Main-Entry-Point des TMS.

#### Felder in der MainTmsSim-Klasse

Das Feld „trackPanelRepository“ hält alle offenen Track-Panels, die dazu dienen für einen einzelnen Zug eine MA beanzutragen.

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Beschreibung |
| static javax.swing.JFrame | [MainFrame](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/MainTmsSim.html#MainFrame) | Das Hauptfenster |
| static java.util.concurrent.Flow.Subscriber<java.lang.String> | [MainSubscriber](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/MainTmsSim.html#MainSubscriber) | Horcht auf Befehl neuzuzeichnen |
| static java.lang.String | [S\_TMS\_ID](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/MainTmsSim.html#S_TMS_ID) | Id dieses TMS |
| static java.util.ArrayList<javax.swing.JPanel> | [trackPanelRepository](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/MainTmsSim.html#trackPanelRepository) | Panels von UIs beantragter MAs |

#### Methoden der MainTmsSim-Klasse

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| static javax.swing.JFrame | [createTmsFrame](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/MainTmsSim.html#createTmsFrame(de.ibw.tms.MainTmsSim.TmsMode))​([MainTmsSim.TmsMode](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/MainTmsSim.TmsMode.html) Mode) | Erstellt das Hauptfenster des TMS |
| static void | [main](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/MainTmsSim.html#main(java.lang.String%5B%5D))​(java.lang.String[] args) | Main-Entry-Point |
| static void | [updateSubViews](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/MainTmsSim.html#updateSubViews())() | Zeichnet Ma-Beantragunsfenster neu. |

Package de.ibw.tms

### Enum MainTmsSim.TmsMode

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<MainTmsSim.TmsMode>, java.lang.constant.Constable

Enclosing class:

[MainTmsSim](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/MainTmsSim.html)

public static enum MainTmsSim.TmsMode

extends java.lang.Enum<MainTmsSim.TmsMode>

Status von Modus des TMS bisher nur Normal.

Package de.ibw.tms

### Class GraphicMoveByMouse

public class GraphicMoveByMouse

extends java.lang.Object

Verwaltet Maus-Clicks in einer Zeichenfläche.

#### Konstruktor des Graphic Move By Mouse

public GraphicMoveByMouse​(javax.swing.JPanel jPanel)

Registriert das Panel um Mouseclicks auf diesem zu verwalten.

Parameters:

jPanel – Jpanel – Dieses Panel wird auf Mouse-Clicks beprüft.

Package de.ibw.tms

### Class LoggingFrame

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

public class LoggingFrame

extends javax.swing.JFrame

Loggt Ausgaben des Benutzers. Diese Fenster wird sofort, nach Konstruktor Aufruf angezeigt.

#### Konstruktor für den Logging-Frame

public LoggingFrame()

Instanziiert das Logging in einem extra Fenster.

Package de.ibw.tms.co

### Class CartesianDialog

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

Direct Known Subclasses:

[SpeedDialog](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/speed/profile/view/SpeedDialog.html)

public abstract class CartesianDialog

extends javax.swing.JDialog

Diese Komponente ist ein modales Dialog-Fenster.

#### Konstruktor für Cartesian-Dialog

public CartesianDialog​(javax.swing.JFrame parentFrame,

java.lang.String sFrameTitle)

Dieser Construktur startet den Modalen Dialog

Parameters:

parentFrame - JFrame - gibt den Frame der wegen der Modalität gesperrt wird, bis der Dialog geschlossen wird.

sFrameTitle - String - Titel des Dialog-Fensters

Package [de.ibw.tms.co](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/co/package-summary.html)

### Class CartesianPanel

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible

Direct Known Subclasses:

[SpeedPanel](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/speed/profile/view/SpeedPanel.html)

public class CartesianPanel

extends javax.swing.JPanel

Diese Komponente ist ein Panel mit Koordinatensystem angepasst von:   
[iberl@verkehrt.tu-darmstadt.de](mailto:iberl@verkehrt.tu-darmstadt.de)

#### Felder des Kartesian-Panels

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Beschreibung |
| static int | [AXIS\_STRING\_DISTANCE](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/co/CartesianPanel.html#AXIS_STRING_DISTANCE) | Gibt an, wie weit die Beschriftung von der Achse entfernt sind. |
| static int | [FIRST\_LENGHT](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/co/CartesianPanel.html#FIRST_LENGHT) | Länge der Achsen-Pfeile. FIRST\_LENGTH zeigt den Anteil des Pfeils auf der Achse dar. SECOND\_LENGTH ist der Abstand von der Achse, die Breite des Pfeiles |
| static int | [SECOND\_LENGHT](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/co/CartesianPanel.html#SECOND_LENGHT) | Abstand des Punktes auf dem Pfeilschenkel zur Achse. |
| static int | [ORIGIN\_COORDINATE\_LENGHT](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/co/CartesianPanel.html#ORIGIN_COORDINATE_LENGHT) | Gibt an, wie dick der Punkt im Ursprung gezeichnet wird. |
| static int | [X\_AXIS\_FIRST\_X\_COORD](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/co/CartesianPanel.html#X_AXIS_FIRST_X_COORD) | Start der x-Achse ab dem linken Panelrand (Das ist auch der Ursprung) |
| static int | [X\_AXIS\_SECOND\_X\_COORD](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/co/CartesianPanel.html#X_AXIS_SECOND_X_COORD) | Ende der X-Achse ab dem linken Panelrand (Das ist der x-Achsenpfeil) |
| static int | [X\_AXIS\_Y\_COORD](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/co/CartesianPanel.html#X_AXIS_Y_COORD) | Abstand der x-Achse vom oberen Panelrand |
| int | [xCoordNumbers](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/co/CartesianPanel.html#xCoordNumbers) | Anzahl der x-Achsenbeschriftungen. |
| static int | [Y\_AXIS\_FIRST\_Y\_COORD](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/co/CartesianPanel.html#Y_AXIS_FIRST_Y_COORD) | Ende der y-Achse gemessen ab dem oberen Panelrand |
| static int | [Y\_AXIS\_SECOND\_Y\_COORD](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/co/CartesianPanel.html#Y_AXIS_SECOND_Y_COORD) | Start der y-Achse gemessen ab dem oberen Panelrand (Ursprung) |
| static int | [Y\_AXIS\_X\_COORD](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/co/CartesianPanel.html#Y_AXIS_X_COORD) | Abstand der y-Achse vom linken Panelrand |
| int | [yCoordNumbers](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/co/CartesianPanel.html#yCoordNumbers) | Anzahl der y-Achsenbeschriftungen. |

#### Methoden des Cartesian-Panels

**public void** paintComponent​(java.awt.Graphics g)

Zeichnet das Koordinatensystem auf dem Panel.

Overrides:

paintComponent in class javax.swing.JComponent

Parameters:

g - Graphics - Java Zeichen Object

**protected void** drawAxis​(java.awt.Graphics2D g2)

Algorthmus der die Achsen zeichnet.

Parameters:

g2 - Graphics2D - 2d Zeichenfunktion

Package [de.ibw.tms.controller](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/controller/package-summary.html)

### Class PositionReportController

All Implemented Interfaces:

IController, java.lang.AutoCloseable, java.util.concurrent.Flow.Publisher

public class PositionReportController

extends java.util.concurrent.SubmissionPublisher

implements IController

Diese Komponente implementiert, was das TMS bei eingehenden Position-Reports unternimmt.

#### Feld im Position-Report-Controller

**public double** D\_DEFAULT\_MIN\_LENGTH

Die Länge des Zuges in Meter, falls das RBC keine Angabe gemacht hat.

#### Methoden des Position-Report-Controller

**public static** [PositionReportController](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/controller/PositionReportController.html) getInstance()

Singelton zu diesem Controller

Returns: PositionReportController - Controller verwaltet PositionReports im TMS

**public void** servePositionReport​(ebd.rbc\_tms.payload.Payload\_14 PositonReport,

java.lang.String sRBC)

Verwaltet eingehenden PositionReport

Parameters:

PositonReport - - Payload\_14 - Informationen des Position-Reports

sRBC - String - Name des RBC

**public void** publish()

Dieser Controller sendet Nachricht an alle Komponenten des TMS, die sich dafür eingeschrieben haben, dass es einen neuen PositionReport gab.

Specified by:

publish in interface IController

**public java.util.List<java.util.concurrent.Flow.Subscriber>** getSubscriberList()

Eine Liste von eingetragenen Empfängern, die Nachricht erhalten, wenn Position-Reports eintreffen. Dazu gehören das Zoom-Fenster, das TMS-Hauptfenster und die Fenster, die zum Erstellen von MAs geöffnet wurden.

Specified by:

getSubscriberList in interface IController

Returns:

List - eine Liste von Empfängern

Package de.ibw.tms.intf

### Class SmartClient

All Implemented Interfaces:

java.lang.Runnable

public class SmartClient

extends RbcModul

Client des TMS zum Server in der SmartLogic

#### Der Konstruktor des Smart-Clients

public SmartClient​(java.lang.String sHost,

int iPort)

Dieser Konstruktor erstellt den Client.

Parameters:

sHost - - Host-ip der SmartLogic.

iPort - - Port des SmartLogic-Host (Smart-Server)

#### Methoden des SmartClients

**public void** start()

Client Startet Verbindung zur SmartLogic

Overrides:

start in class RbcModul

**public void** run()

Konkretes Verhalten beim Start des Clients. Benutzt den Smart-ClientHandler, um Nachrichten an die SL zu senden.

Specified by:

run in interface java.lang.Runnable

Overrides:

run in class RbcModul

Package de.ibw.tms.intf

### Class SmartClientHandler

All Implemented Interfaces:

io.netty.channel.ChannelHandler, io.netty.channel.ChannelInboundHandler

public class SmartClientHandler

extends io.netty.channel.SimpleChannelInboundHandler<[SmartServerMessage](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/smart/logic/intf/messages/SmartServerMessage.html)>

Client-Handler im TMS. Er behandelt Interaktionen des TMS mit der SL und umgekehrt. Beides wird in dieser Klasse nur für die TMS Seite definiert.

#### Methoden des Smart-Client-Handlers

**public static SmartClientHandler** getInstance()

Singelton, um den Client-Handler des TMS widerzugeben

Returns:

SmartClientHandler - Handler der Interaktion im TMS

**public void** channelActive​(io.netty.channel.ChannelHandlerContext channelHandlerContext)

throws java.lang.Exception

Definiert was das TMS unternimmt, solange eine Verbindung zur SL aktiv ist. Bis jetzt wird konkret der Postausgang des TMS in den Channel gegeben.

Specified by:

channelActive in interface io.netty.channel.ChannelInboundHandler

Overrides:

channelActive in class io.netty.channel.ChannelInboundHandlerAdapter

Parameters:

channelHandlerContext - - Netty-Context

Throws:

java.lang.Exception - - Fehler bei der Kommunikation zur SL

**public void** exceptionCaught​(io.netty.channel.ChannelHandlerContext channelHandlerContext,

java.lang.Throwable cause)

Verhalten wenn Fehler zur SL Verbindung auftreten. Bisher wird die Kommunikation nicht wiederaufgebaut.

Specified by:

exceptionCaught in interface io.netty.channel.ChannelHandler

Specified by:

exceptionCaught in interface io.netty.channel.ChannelInboundHandler

Overrides:

exceptionCaught in class io.netty.channel.ChannelInboundHandlerAdapter

Parameters:

channelHandlerContext - - Netty-Context

cause - - Fehler der auftrat

**protected void** channelRead0​(io.netty.channel.ChannelHandlerContext channelHandlerContext, SmartServerMessage smartServerMessage)

throws java.lang.Exception

Definiert, was das TMS unternimmt, wenn eine Nachricht der SL eintrifft. Bisher wird hier unterschieden ob die SL eine Nachricht des RBC weitergegeben hat oder eine Nachricht der SL eintrifft, weil eine MA akzeptiert oder verworfen wurde.

Specified by:

channelRead0 in class io.netty.channel.SimpleChannelInboundHandler<SmartServerMessage>

Parameters:

channelHandlerContext - - Netty-Context

smartServerMessage - SmartServerMessage - Nachricht von der SL

Throws:

java.lang.Exception - - Fehler in der Kommunikation zur SL

**public void** sendCommand​(TmsMessage TmsCmd)

throws ebd.rbc\_tms.util.exception.MissingInformationException

Diese Methode gibt die TMS-Nachricht in den Postausgang an die SL.

Parameters:

TmsCmd - TmsMessage - Nachricht die an die SL gesendet werden soll

Throws:

ebd.rbc\_tms.util.exception.MissingInformationException

Package de.ibw.tms.intf

### Class TmsMessage

All Implemented Interfaces:

java.io.Serializable

Direct Known Subclasses:

[TmsMovementAuthority](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/TmsMovementAuthority.html)

public class TmsMessage

extends ebd.rbc\_tms.Message<[Commands](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/Commands.html)>

implements java.io.Serializable

Wrapper-Klasse zu einem Command. Objekte dieser Klasse werden als JSON gesendet. Es werden alle Exposed Felder in das JSON eingebettet. Darunter sind auch Felder der Vaterklassen, die geerbt wurden, enthalten. Diese Objekte werden eingesetzt, um Nachrichten zur SL zu senden.

#### Felder der Tms-Messages

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Beschreibung |
| java.lang.String | [rbc\_id](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/TmsMessage.html#rbc_id) | Ziel RBC-Id |
| java.lang.String | [tms\_id](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/TmsMessage.html#tms_id) | Eigene TMS-Id |
| int | [type](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/TmsMessage.html#type) | Typ der Nachricht generell 0 |

#### Konstruktor für TMS-Messages

protected TmsMessage​(java.lang.String tms\_id,

java.lang.String rbc\_id,

Commands payload)

Generelle Konstruktor-Definition für Kind-Klassen.

Parameters:

tms\_id - String - Id des angefragten TMS

rbc\_id - String - Id des angefragten RBC

payload - - Commands - Hat einen Payload eines TMS-Commands

Package de.ibw.tms.intf

### Class TmsMovementAuthority

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<TmsMessage>

public class TmsMovementAuthority

extends TmsMessage

implements java.lang.Comparable<TmsMessage>

Nachricht an die SL diese Ma-Nachricht zu prüfen.

#### Feld in der TMS-Movement-Authority

@Expose

**public CheckMovementAuthority** payload

Nachrichteninhalt, Expose markiert, dass es als JSON serialisiert wird.

#### Konstruktor einer Tms-Movement-Authority

public TmsMovementAuthority​(java.lang.String tms\_id,

java.lang.String rbc\_id,

CheckMovementAuthority payload)

Konstruktor einer TMS-Ma-Nachricht an die SL

Parameters:

tms\_id - - eigene Id des TMS

rbc\_id - - Ziel-RBC-Id

payload - - Inhalt der Nachricht

#### Methoden der TMS-Movement-Authority

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| int | [compareTo](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/TmsMovementAuthority.html#compareTo(de.ibw.tms.intf.TmsMessage))​([TmsMessage](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/TmsMessage.html) otherMessage) | Vergleich von Priority von Nachrichten. (Niedrig zuerst) |
| [Commands](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/Commands.html) | [getPayload](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/TmsMovementAuthority.html#getPayload())() | Holt aus dieser Nachricht den Nachrichteninhalt |
| java.lang.String | [toString](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/TmsMovementAuthority.html#toString())() | String-Darstellung dieser Nachricht |

Package de.ibw.tms.intf.cmd

### Class CheckMovementAuthority

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<Commands>

public class CheckMovementAuthority

extends Commands

Dieser Befehl entsteht im TMS und weist die SL die beinhaltete MA zu prüfen.

#### Felder des Check-Movement-Authority

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Beschreibung |
| java.lang.Long | [lPriority](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/CheckMovementAuthority.html#lPriority) | Priority dieser Nachricht im Postausgang des TMS |
| [RbcMaAdapter](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/ma/RbcMaAdapter.html) | [MaAdapter](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/CheckMovementAuthority.html#MaAdapter) | Nachricht an das RBC |
| [MaRequestWrapper](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/ma/MaRequestWrapper.html) | [MaRequest](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/CheckMovementAuthority.html#MaRequest) | Ma to check |
| java.lang.String | [rbc\_id](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/CheckMovementAuthority.html#rbc_id) | Id des RBC |
| java.lang.String | [tms\_id](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/CheckMovementAuthority.html#tms_id) | Id des TMS |
| java.util.UUID | [uuid](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/CheckMovementAuthority.html#uuid) | Kommunikations UUid |

#### Konstruktor einer Check-Movement-Authority

public CheckMovementAuthority​(long lPriority)

Dieser Konstruktor erstellt einen neuen leeren Check-Befehl mit einer Priority

Parameters:

lPriority - long - Priority im TMS Postausgang

#### Methoden der Check-Movement-Authority

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| boolean | [equals](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/CheckMovementAuthority.html#equals(java.lang.Object))​(java.lang.Object o) | Vergleich ob dieser Befehl und o derselbe sei. |
| static [CheckMovementAuthority](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/CheckMovementAuthority.html) | [getDummyMovementAuthorityCommand](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/CheckMovementAuthority.html#getDummyMovementAuthorityCommand())() |  |
| int | [hashCode](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/CheckMovementAuthority.html#hashCode())() | Hashcode dieser Nachricht |
| static void | [main](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/CheckMovementAuthority.html#main(java.lang.String%5B%5D))​(java.lang.String[] args) |  |
| java.lang.String | [parseToJson](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/CheckMovementAuthority.html#parseToJson())() | Stellt den Befehl als JSON dar. |
| java.lang.String | [toString](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/CheckMovementAuthority.html#toString())() | Std String Widergabe |

Package de.ibw.tms.intf.cmd

### Class Commands

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<Commands>

Direct Known Subclasses:

CheckMovementAuthority

public class Commands

extends ebd.rbc\_tms.Payload

implements java.lang.Comparable<Commands>, java.io.Serializable

Diese Klasse stellt einen Befehl in allgemeiner Form vom TMS an die SL dar. Bisher gibt es nur CheckMovementAuthority.

#### Felder von Commands

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Beschreibung |
| java.lang.String | [CommandType](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/Commands.html#CommandType) | String Typ der Nachricht der auch gesendet wird, damit die SL den Typ des Befehls kennt. |
| java.lang.Long | [lPriority](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/Commands.html#lPriority) | Priority wird aber in dem konkret vererbten Befehl definiert |
| static java.lang.String | [S\_CHECK\_MOVEMENT\_AUTHORITY](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/Commands.html#S_CHECK_MOVEMENT_AUTHORITY) | String-Id der CheckMovementAuthority. |

#### Konstruktor von Commands

Commands​(long lPriority)

Erstellt einen Aufruf für Unterklassen bereit.

Parameters:

lPriority - long - Priority dieses Befehls.

#### Methoden von Commands

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| int | [compareTo](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/Commands.html#compareTo(de.ibw.tms.intf.cmd.Commands))​([Commands](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/Commands.html) commands) | Vergleicht zwei Commands und die niedrigere Priority erhält den Vorrang |
| static java.lang.Class | [getClassByString](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/Commands.html#getClassByString(java.lang.String))​(java.lang.String sType) | Gibt die Klasse zur String Id wieder. |
| static java.lang.reflect.Type | [getTypeByString](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/intf/cmd/Commands.html#getTypeByString(java.lang.String))​(java.lang.String sType) | Gibt den Typ zur String Id wieder. |

Package de.ibw.tms.plan\_pro.adapter

### Class CrossingSwitch

All Implemented Interfaces:

ICompareTrackMeter

public class CrossingSwitch

extends java.lang.Object

implements ICompareTrackMeter

Ein Wrapper für Weichen aus PlanPro.

#### Konstruktor von Crossing-Switches

public CrossingSwitch​(CWKrAnlage A,

CWKrGspElement E,

CWKrGspKomponente C)

throws java.text.ParseException

Instanziiert eine Weichenwrappter

Parameters:

A - CWKrAnlage - Weicheanlagedaten

E - CWKrGspElement - Weichenelement

C - CWKrGspKomponente - Weichenkomponente

Throws:

java.text.ParseException – falls, die Streckenkilometrierung sich nicht in double konvertieren lässt.

#### Methoden von Crossing-Switches

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| CWKrAnlage | getAnlage() | Gibt Weichenanlagendaten wieder |
| CWKrGspKomponente | getComponent() | Gibt Weichenkomponente wieder |
| CWKrGspElement | getElement() | Gibt Weichenelement wieder |
| java.util.ArrayList<java.lang.String> | getSupportedTracks() | Gibt eine Liste von Strecken-Ids von Strecken, die eine Kilometrierung auf dieser Weiche haben. |
| java.math.BigDecimal | getTrackMeterByTrackId​(java.lang.String trackId) | Gibt die Streckenkilometrierung einer Strecke wieder. |

Package de.ibw.tms.plan\_pro.adapter

### Enum PlanProTmsAdapter.PlanProVersion

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<PlanProTmsAdapter.PlanProVersion>, java.lang.constant.Constable

Enclosing class:

PlanProTmsAdapter

public static enum PlanProTmsAdapter.PlanProVersion

extends java.lang.Enum<PlanProTmsAdapter.PlanProVersion>

Verwaltet unterschiedliche PlanProVersionen in den Konstanten dieses Enums

#### Enum-Konstanten

V1\_9\_0\_PATCHED – Erste Version in der eine Implementierung möglich wurde.

Package de.ibw.tms.plan\_pro.adapter.topology

### Class TopologyGraph

public class TopologyGraph

extends java.lang.Object

Der Graph aus Knoten und Kanten der Topologie.

Package de.ibw.tms.plan\_pro.adapter.topology

#### Sub-Class TopologyGraph.Edge

All Implemented Interfaces:

IConnectable, IViewNaming, java.io.Serializable

Enclosing class:

TopologyGraph

public static class TopologyGraph.Edge

extends [TrackElement](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/ma/physical/TrackElement.html)

Topologische Kante

##### Felder von Topologygraph-Edge

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Beschreibung |
| TopologyGraph.Node | **A** | Knoten A der Kante |
| TopologyGraph.Node | **B** | Knoten B dieser Kante |
| double | **dTopLength** | Kantenlänge in Meter |
| java.lang.String | **sId** | PlanPro KantenId |
| TopologyConnect | **TopConnectFromA** | Verbindungsart des Knoten A (Spitze, Rechts, Links) |
| TopologyConnect | **TopConnectFromB** | Verbindungsart des Knoten B (Spitze, Rechts, Links) |

##### Konstruktor der Topologygraph-Edge

public Edge​(TopologyGraph.Node A,

TopologyConnect topConnectFromA,

TopologyGraph.Node B,

TopologyConnect topConnectFromB,

CTOPKante Edge)

Dieser Konstruktor instanziiert eine Topologische Kante

Parameters:

A - TopologyGraph.Node - Topologischer Knoten A

topConnectFromA - TopologyConnect - Verbindungsart an A (Rechts, Links, Spitze)

B - TopologyGraph.Node - Topologischer Knoten B

topConnectFromB - TopologyConnect - Verbindungsart an B (Rechts, Links, Spitze)

Edge - - CTOPKante - PlanPro-Modell dieser Kante

##### Methoden der Topologygraph-Edge

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| java.util.ArrayList<[CGEOKante](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/plan_pro/modell/geodaten/_1_9_0/CGEOKante.html)> | [getPaintListGeo](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan_pro/adapter/topology/TopologyGraph.Edge.html#getPaintListGeo())() | Geographische Sub-Kanten dieser topologischen Kante |
| [CTOPKante](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/plan_pro/modell/geodaten/_1_9_0/CTOPKante.html) | [getPlanProEdge](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan_pro/adapter/topology/TopologyGraph.Edge.html#getPlanProEdge())() | Gibt die Topologische Kante nach Definition in PlanPro wieder |
| [Rail](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/Rail.html) | [getRail](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan_pro/adapter/topology/TopologyGraph.Edge.html#getRail())() | Gleis-Model (geographisch) dieser Kante. |
| java.lang.String | [getViewName](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan_pro/adapter/topology/TopologyGraph.Edge.html#getViewName())() | Bezeichnung der Kante in Ansichten |
| java.lang.Boolean | [isFromNodeAtoNodeBisInTrackDirection](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan_pro/adapter/topology/TopologyGraph.Edge.html#isFromNodeAtoNodeBisInTrackDirection())() | Diese Methode gibt an ob von Knoten A zum Knoten B dieser Kante in Streckenkilometrierung verläuft. |
| void | setRail​(Rail r) | Setzt das logische-geographische Gleis dieser topologischen Kante |

**public void** setPaintListGeo​(java.util.ArrayList<CGEOKante> paintListGeo)

Setzt die Geographische Kante dieser Topologischen Kante

Parameters:

paintListGeo - - List - Geo-Kantenliste

**public boolean** equals​(java.lang.Object obj)

Diese Methode vergleicht zwei Kanten, ob sie identisch sind.

Overrides:

equals in class java.lang.Object

Parameters:

obj - - Object - Kantenobjektt zum Vergleich mit dieser Kante

Returns:

boolean - gibt an ob die Kanten identisch sind (true) sonst (false)

**public boolean** checkIfStartAndEndpointBelongsToThisEdge​(java.lang.String sIdTopNode1,java.lang.String sIdTopNode2)

Diese Methode untersucht ob zwei Knoten-Ids der Knoten dieser Kante angehören.

Parameters:

sIdTopNode1 - String - Knoten-Id 1

sIdTopNode2 - String - Knoten-Id 2

Returns:

boolean - gibt an ob die Knoten dieser Kante teilhaft sind (true) sonst (false)

Package de.ibw.tms.plan\_pro.adapter.topology

#### Sub-Class TopologyGraph.Node

All Implemented Interfaces:

IConnectable, IViewNaming, java.io.Serializable

Enclosing class:

TopologyGraph

public static class TopologyGraph.Node

extends TrackElement

Topologischer Knoten

##### Felder von Topology-Graph-Knoten

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Beschreibung |
| java.util.HashSet<TopologyGraph.Edge> | inEdges | Kanten eingehend |
| java.lang.String | name | Knoten Bezeichnung |
| java.lang.Object | NodeImpl | Knoten-Realisierung |
| java.lang.Class | NodeType | Klasse des Knoten |
| java.util.HashSet<TopologyGraph.Edge> | outEdges | Kanten ausgehend |
| java.lang.String | TopNodeId | PlanPro Knoten Id |

##### Konstruktor der Topologygraph-Knoten

public Node​(java.lang.String name,

java.lang.String topNodeId,

GeoCoordinates GeoCo)

Konstruktor zur Instanziierung eines Knoten im Topologie-Graphen.

Parameters:

name - String - Bezeichnung des Knoten

topNodeId - String - PlanPro Id des Knoten

GeoCo - GeoCoordinates - Koordinaten des Knotens

##### Methoden der Topologygraph-Knoten

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| [CrossoverModel](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/CrossoverModel.html) | [getModel](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan_pro/adapter/topology/TopologyGraph.Node.html#getModel())() | Diese Methode zieht die Vermittlung zur Weiche |
| java.lang.String | [getViewName](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan_pro/adapter/topology/TopologyGraph.Node.html#getViewName())() | Knotenname in Ansichten |
| java.lang.String | [toString](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan_pro/adapter/topology/TopologyGraph.Node.html#toString())() | Knotenbezeichnung als String |

public TopologyGraph.Node addEdge​(TopologyGraph.Node Node,

TopologyGraph.Edge E)

Generiert eine Kante diesem und einem weiteren Knoten

Parameters:

Node - TopologyGraph.Node - Weiterer Knoten zur Kante

E - TopologyGraph.Edge - Kante aus beiden Knoten

Returns:

Node - diesen Knoten (des Methodenaufrufs)

#### Felder eines Topologie-Graphen

public static [DefaultRepo](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/util/DefaultRepo.html)<java.lang.String,​[DefaultRepo](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/util/DefaultRepo.html)<java.lang.String,​[TopologyGraph.Edge](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan_pro/adapter/topology/TopologyGraph.Edge.html)>> twoTopPointBelongsToEdgeRepo

Repository, dass eine Topoologisch Kante TopologyGraph.Edge über die String-PlanPro-Id der Knoten speichert.

public java.util.HashMap<java.lang.String,​[TopologyGraph.Edge](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan_pro/adapter/topology/TopologyGraph.Edge.html)> EdgeRepo

HashMap die für die PlanPro-Kanten-ID die Topologische Kante speichert

public static java.util.HashMap<java.lang.String,​TopologyGraph.Node> NodeRepo

HashMap die für die PlanPro-Knoten-ID einen Topologische Knoten speichert

#### Methoden des Topologie-Graphen

Zur Suche in einem Graphen wird ein Startknoten gesetzt. Von dort wird aus wird der Graph Stück für Stück zusammengesetzt. Seit kurzem wird von einer Hauptstrecke ausgehend eine Kante mit einem Knoten gewählt, der ein Ende hat. Die Streckenorientierung hat auch den Vorteil, dass die Kilometrierung besser programmiert werden kann.

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| static TopologyGraph.Node | getLeftmostNode() | Holen des Startknotens |
| static java.lang.Double | getXofCurrentLeftMostNode() | Holt die X-coordinate des Startknotens |
| static void | main​(java.lang.String[] args) |  |
| static void | setLeftmostNode​(TopologyGraph.Node leftmostNode) | Setzen des Startknotens |

Package de.ibw.tms.plan\_pro.adapter.topology.trackbased

### Class TopologyFactory

All Implemented Interfaces:

ITopologyFactory

public class TopologyFactory

extends java.lang.Object

implements ITopologyFactory

Diese Klasse dient zur Erzeugung eines Topologischen Graphen

#### Feld in der Topologie-Factory

public static boolean shallAssignToActivePlanData

Dieses Feld gibt an, ob der erzeugte Graph in PlanData gespeichert werden soll.

#### Methoden der Topologie-Factory

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| TopologyGraph | connectTopology() | Diese Methode generiert den TopologyGraph |
| java.util.List<Balise> | getBalises() | List - gibt eine Liste von Balisen aus der PlanProDatei wieder |
| DefaultRepo<java.lang.String,​CStrecke> | getTrackRepo() | Gibt eine Repository wieder, das die Strecken aus PlanPro mithilfe der Plan-Pro-Schlüssel-ID abrufbar macht. |

**public DefaultRepo<java.lang.Class<?>,​DefaultRepo<java.lang.String,​CBasisObjekt>>** getGeoBundle()

Specified by:

getGeoBundle in interface ITopologyFactory

Diese Methode gibt ein Bundle von Repositorien, die Geo-Daten mit deren Klassen darstellen. Ein Bundle wird selbst durch ein Repositorium implementiert. Als Bundle-Schlüssel wird eine der folgenden Geographische-Klasse erwartet:

CGEOPunkt.class, CGEOKnoten.class, CGEOKante.class, CTOPKante.class, CTOPKnoten.class

Im Geo-Bundle sind auch die Topologischen Klassen CTOPKante und CTOPKnoten enthalten. Das hat den Vorteil, dass man kein zweites Bundle verwenden muss.

Ruft man eine Klasse aus dem Bundle mit getModel() ab, erhält man eine anderes Repositorium.

Das andere innere Repositorium enthält das Objekt zu den jeweiligen obigen Klassen über deren PlanPro-String-Id-Schlüssel.

Ein Möglicher Aufruf wäre:

CGEOKante K123 = (CGEOKante) this.getGeoBundle().getModel(CGEOKante.class).getModel(„123-ABC“);

Die Objekte sind als CBasisObjekt abrufbar. Deswegen muss man auf die jeweilige Subklasse einen Cast vornehmen.

**public**[**DefaultRepo**](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/util/DefaultRepo.html)**<java.lang.String,​CTOPKnoten>** getGeoNodeToTopNodeRepo()

Diese Getter-Methode gibt ein Repository zurück, das eine die String-Id eines Geo-Knoten als key besitzt. Als Wertebereich werden Topologische Knoten zurückgegeben.

Specified by:

getGeoNodeToTopNodeRepo in interface ITopologyFactory

Returns:

Ein Repository mit Geo-Key und Value einen CTOPKnoten

**public DefaultRepo<java.lang.String,​CGEOKnoten>** getTopNodeToGeoNodeRepo()

Diese Getter Methode gibt ein Repository zurück, das die String-Id eines Top-Knoten als key besitzt. Als Wertebereich werden Geographisch Knoten zurückgegeben.

Specified by:

getTopNodeToGeoNodeRepo in interface ITopologyFactory

Returns:

Ein Repository mit top-Key und Value eine CGEOKnoten

**public void** setTopLines​(java.util.List<CTOPKante> topLines)

Diese Methode ist derzeit unbenutzt, weil die CTOPKante-Liste direkt aus Plan-Pro gesetzt wird, ohne diese Methode aufzurufen.

Specified by:

setTopLines in interface ITopologyFactory

**public void** setGeoBundle​(DefaultRepo<java.lang.Class<?>,​DefaultRepo<java.lang.String,​CBasisObjekt>> geoBundle)

Diese Methode ist derzeit unbenutzt, weil das Geo-Bundle beim Verarbeiten der Plan-Pro-Datei, gesetzt wird, ohne diese Methode aufzurufen.

Specified by:

setGeoBundle in interface ITopologyFactory

**public void** setGeoNodeToGeoEdgesRepo​([DefaultRepo](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/util/DefaultRepo.html)<java.lang.String,​java.util.List<[CGEOKante](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/plan_pro/modell/geodaten/_1_9_0/CGEOKante.html)>> geoNodeToGeoEdgesRepo)

Diese Methode ist derzeit unbenutzt.

**public void** handleBranchingPoints()

throws java.text.ParseException

Diese Methode generiert Weichen CrossingSwitch und speichert sie in eine Liste in PlanData

Specified by:

handleBranchingPoints in interface ITopologyFactory

Throws:

java.text.ParseException - - Wenn undefinierte Werte in Plan-Pro vorhanden sind, diese aber benötigt werden.

**public void** mapBalisesToCoordinate()

Diese Methode ordnet den Balisen Koordinaten zu.

Specified by:

mapBalisesToCoordinate in interface ITopologyFactory

Package de.ibw.tms.plan.elements

### Class BranchingSwitch

All Implemented Interfaces:

IConnectable, ICrossover, Iinteractable, ITrack, IViewNaming, java.awt.Shape, java.io.Serializable, java.lang.Cloneable, java.util.concurrent.Flow.Subscriber<CrossoverMainModel>

public class BranchingSwitch

extends java.awt.geom.Point2D.Double

implements java.awt.Shape, ICrossover, ITrack, java.util.concurrent.Flow.Subscriber<CrossoverMainModel>

Branching Switch ein Modell einer Weiche Geographischer und Logischer Art

#### Enum BranchingSwitch.CrossoverStatus

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<BranchingSwitch.CrossoverStatus>, java.lang.constant.Constable

Enclosing class:

BranchingSwitch

public static enum BranchingSwitch.CrossoverStatus

extends java.lang.Enum<[BranchingSwitch.CrossoverStatus](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/BranchingSwitch.CrossoverStatus.html)>

Status, die eine Weiche haben kann

|  |  |
| --- | --- |
| Enum Konstante | Beschreibung |
| [BUSY](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/BranchingSwitch.CrossoverStatus.html#BUSY) | Weiche im Stellprozess |
| [LEFT](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/BranchingSwitch.CrossoverStatus.html#LEFT) | Weiche in Linkslage |
| [RIGHT](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/BranchingSwitch.CrossoverStatus.html#RIGHT) | Weiche in Rechtslage |
| [UNSAFE](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/BranchingSwitch.CrossoverStatus.html#UNSAFE) | Weiche nicht eingerastet |
| [UNSAFE\_LEFT](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/BranchingSwitch.CrossoverStatus.html#UNSAFE_LEFT) | Weiche links, aber nicht eingerastet |
| [UNSAFE\_RIGHT](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/BranchingSwitch.CrossoverStatus.html#UNSAFE_RIGHT) | Weiche rechts, aber nicht eingerastet |

#### Enum BranchingSwitch.ViewType

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<BranchingSwitch.ViewType>, java.lang.constant.Constable

Enclosing class:

BranchingSwitch

public static enum BranchingSwitch.ViewType

extends java.lang.Enum<BranchingSwitch.ViewType>

Enum, das definiert welches Bild für die Weiche benutzt werden soll

|  |  |
| --- | --- |
| Enum Konstante | Beschreibung |
| Branch\_LRU |  |
| Branch\_ORL |  |
| Branch\_RLO |  |
| Branch\_ULR |  |

#### Methoden eines Branching-Switches

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| java.awt.Rectangle | getBounds() | Gibt einen Recheckbereich der GUI-Klickbaren-Größe der Weiche an. |
| java.awt.geom.Rectangle2D | getBounds2D() | Gibt einen 2-D-Recheckbereich der GUI-Clickbaren-Größe der Weiche an. |
| SlipConnectionPoint | getBranchingPoint() | Gibt den Schlupf der Weiche wieder. |
| CrossoverController | getController() | Gibt Controller, der das Kontextmenu, bei Userinteraktion, nach Auswahl, die entsprechende Aktion ausführt, wieder |
| java.awt.image.BufferedImage | getImage() | Bild das die Weiche darstellt. |
| BranchingSwitch.CrossoverStatus | getStatus() | Hold den aktuellen Status der Weiche |
| TrackElement | getTrackReference() | Gibt Schlupf der Weiche wieder |
| java.util.ArrayList<javax.swing.JComponent> | getViewElements() | Definiert die Ui-Elemente im Kontextmenu der Weiche |
| java.lang.String | getViewName() | Bezeichnung der Weiche im Menu und auch auf dem Track Plan |

**public void** setPeekRail​(Rail PeekRail)

Setzt das Gleis an der Weichenspitze

Parameters:

PeekRail - Rail - Gleis der Weichenspitze, das gesetzt wird

**public void**setNode​(TopologyGraph.Node node)

Setzt Bezug zum Topologischen Knoten-Kanten-Listen-Modell

Parameters:

node - TopologyGraph.Node - Der Topologische Knoten für diese Weiche

**public void** onSubscribe​(java.util.concurrent.Flow.Subscription subscription)

Weichenmodel-Änderungen dieser Weiche, werden hier als bewacht angemeldet. Der Branching-Switch erhält über diese Subscription, Änderung dessen Models.

Specified by:

onSubscribe in interface java.util.concurrent.Flow.Subscriber<CrossoverMainModel>

Parameters:

subscription - - Anmeldungsobject bei Änderungen des Models

**public void** onNext​(CrossoverMainModel item)

Ein Weichenmodel wurde geändert.

Specified by:

onNext in interface java.util.concurrent.Flow.Subscriber<CrossoverMainModel>

Parameters:

item - CrossoverMainModel Neues Modell der Weiche, das gespeichert wird.

**public void** onError​(java.lang.Throwable throwable)

Neues Weichenmodel wirft Fehler. Bisher wird nur der Fehler in der Konsole angezeigt

Specified by:

onError in interface java.util.concurrent.Flow.Subscriber<CrossoverMainModel>

Parameters:

throwable - - Fehler der geworfen wird

**public void** onComplete()

Neues Weichenmodel wurde in onNext behandelt. Hier wird beschrieben was dann passiert. Bisher keine Folgeaktionen.

Specified by:

onComplete in interface java.util.concurrent.Flow.Subscriber<[CrossoverMainModel](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/model/CrossoverMainModel.html)>

**public static**[**BranchingSwitch**](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/BranchingSwitch.html) createCrossover​(java.lang.Float fLinkTime, SingleSlip BranchPoint, double dx, double dy, java.lang.String sName, BranchingSwitch.ViewType ViewType)

Factory Methode, das ein Weichenmodell einer einzelnen Weiche generiert.

Parameters:

fLinkTime - - Float - Dauer der Weichenstellung in ms

BranchPoint - - SingleSlip - Schlupf der Weiche

dx - double - X-Position auf geographischen Fenstern

dy - double - Y-Position auf geographischen Fenstern

sName - String - Name der Weiche

ViewType - BranchingSwitch.ViewType - Bild der Weiche

Returns:

BranchingSwitch - Gibt Weichenmodell mit den Parametern wieder.

**public void** setfLinkageTimeInMs​(float fLinkageTimeInMs)

Setzt die Umschaltdauer des Weichenmodells

Specified by:

setfLinkageTimeInMs in interface ICrossover

Parameters:

fLinkageTimeInMs - float - Umschaltzeit in ms

**public void** setStatus​(BranchingSwitch.CrossoverStatus status)

Setzt den Weichenstatus.

Specified by:

setStatus in interface ICrossover

Parameters:

status - - BranchingSwitch.CrossoverStatus der neue Status der Weiche.

Package de.ibw.tms.plan.elements

### Class CrossoverModel

public class CrossoverModel

extends java.lang.Object

Branching Switch ein Modell einer Weiche Logischer Art. Verbindet das Geographische Modell der Weiche BranchingSwitch mit dem Topologischen TopologyGraph.Node

#### Felder des Crossover-Modells

**public static**[**DefaultRepo**](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/util/DefaultRepo.html)**<**[**TopologyGraph.Node**](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan_pro/adapter/topology/TopologyGraph.Node.html)**,​**[**CrossoverModel**](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/CrossoverModel.html)**>** CrossoverRepo

Ein Repository das einen topologischen Knoten als key hat und diese Vermittlungs-Klasse CrossoverModel als Value wiedergibt.

**public static**[**DefaultRepo**](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/util/DefaultRepo.html)**<**[**ControlledTrackElement**](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/ma/physical/ControlledTrackElement.html)**,​**[**CrossoverModel**](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/CrossoverModel.html)**>** BranchToCrossoverModelRepo

Ein Repository das den Schlupf als key hat und diese diese Vermittlungs-Klasse CrossoverModel als Value widergibt.

#### Methoden des Crossover-Modells

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| void | createPositionedRelation() | Verlinkt die Gleise mit dieser Weiche |
| TopologyGraph.Node | getNode() | Gibt die Topologische Weiche wieder |
| SingleSlip | getRailWaySlip() | Gibt den Schlupf der Weiche wieder |

**public static void** createCrossoverModel​(TopologyGraph.Node node,

TopologyConnect topConnect,

SingleSlip railWaySlip,

BranchingSwitch railWaySwitch)

Factory Method dieser Vermittlung

Parameters:

node - TopologyGraph.Node - Topologische Knoten

topConnect - - [TopologyConnect](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan_pro/adapter/topology/TopologyConnect.html) Anschlussart an diese Weiche - Rechts-Links oder Spitze

railWaySlip - SingleSlip - Schlupf dieser weiche

railWaySwitch - BranchingSwitch - Geographische Model der Weiche

Package de.ibw.tms.plan.elements

### Class Rail

All Implemented Interfaces:

Iinteractable, ITrack, IViewNaming, java.awt.Shape, java.io.Serializable, java.lang.Cloneable

public class Rail

extends java.awt.geom.Line2D.Double

implements Iinteractable, ITrack

Geographisches Gleis. Stellt eine Topologie-Kante als Luftlinie dar und macht diese interagierbar.

#### Felder von Rail

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Beschreibung |
| IConnectable | ConA | Verbindung eines TrackElements am Ende A |
| IConnectable | ConB | Verbindung eines TrackElements am Ende B |
| static double | dRailTolerance | Click-Abstand dieses Gleises zum Klickpunkt |
| int | iStroke | Zeichendicke der Gleiskante |
| java.lang.String | segmentName | Name des Gleissegmentes |

#### Konstruktor von Rail

public Rail​(double x1,

double y1,

double x2,

double y2,

java.util.List<Rail> addTo,

IConnectable IConA,

IConnectable IConB,

Chainage Cb,

Chainage Cc,

ApplicationDirection Navigal,

int vmax,

ApplicationDirection Direction,

TrackElementStatus Status)

Dieser Konstruktur erstellt ein Geographisches Gleis

Parameters:

x1 - - xPosition 1

y1 - - yPosition 1

x2 - - xPosition 2

y2 - - yPosition 2

addTo - List - Liste von Gleisen, die diese Gleis zugeordnet werden, nach Erstellung

IConA - TrackElement - Anschluss A

IConB - TrackElement - Anschluss B

Cb - - not used

Cc - - not used

Navigal - - ApplicationDirection von A zu B, oder B zu A, oder beides

vmax - int - maximale Geschwindigkeit auf diesem Gleis

Direction - - ApplicationDirection Streckenrichtung von A zu B oder B zu A

#### Methoden von Rail

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| TopologyGraph.Edge | getEdge() | Gibt die Topologische Kante dieses Gleis wieder |
| TrackElement | getTrackReference() | Gibt Trail als Logik wieder |
| Trail | getTrailModel() | Gibt logisches Model des Gleises wieder |
| java.util.ArrayList<javax.swing.JComponent> | getViewElements() | Kontextmenu-Elemente bei Rechtsklick am Gleis |
| java.lang.String | getViewName() | Bezeichnung des Gleises |
| void | setEdge​(TopologyGraph.Edge edge) | Setzt Topologische Kante zu dem Gleis |

Package de.ibw.tms.plan.elements

### Class UiTools

public class UiTools

extends java.lang.Object

Liest Bild aus den Ressourcen der .jar-Executable.

#### Methode von Ui-Tools

**public static java.awt.image.BufferedImage** handleImaging​(java.lang.ClassLoader cl,

java.lang.String s)

throws java.io.IOException

Gibt angegebenes Bild wieder.

Parameters:

cl - ClassLoader kann vom Klassenattribut bezogen werden

s - String - Bezeichnung in den Ressourcen

Returns:

BufferedImage - Bild als Ergebnis oder null wenn nicht gefunden

Throws:

java.io.IOException - - Falls eine Datei nicht zugreifbar ist

Package de.ibw.tms.plan.elements.model

### Class CrossoverEnumModel

public class CrossoverEnumModel

extends EnumModel

Dieses Enum-Model beschreibt die Lage einer Weiche mit deren Zungen.

#### Methode des Crossover-Enum-Model

**public**[**EnumModel.EnumField**](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/trackplan/EnumModel.EnumField.html)**[]** getEnumMappingList()

Inhalte dieses EnumModels

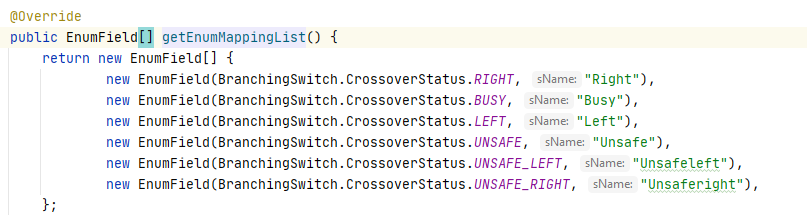
Specified by:

getEnumMappingList in class EnumModel

Returns:

EnumField[] - mögliche Status einer Weiche

Das Crossover-Enum-Modell umhüllt Enum-Felder mit einem Namen.



Package de.ibw.tms.plan.elements.model

### Class CrossoverMainModel

public class CrossoverMainModel

extends java.lang.Object

Status einer Weiche.

#### Felder des Crossover-Main-Modell

**public**[**BranchingSwitch.CrossoverStatus**](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/BranchingSwitch.CrossoverStatus.html) CrossoverStatus

Status der Weichenlage.

**public float** fLinkageTimeInMs

Dauer der Umschaltzeit der Weiche beim Wechsel einer Weichenlage.

Package de.ibw.tms.plan.elements.model

### Class PlanData

All Implemented Interfaces:

java.util.concurrent.Flow.Subscriber<GradientProfile>

public class PlanData

extends java.lang.Object

implements java.util.concurrent.Flow.Subscriber<GradientProfile>

Daten Topographischer und Geographischer Art. Hier wird in Zukunft auch das Gradienten-Profil abgebildet.

Package de.ibw.tms.plan.elements.model

#### Sub-Class PlanData.TrackElementPositionCalc

Enclosing class:

PlanData

public static class PlanData.TrackElementPositionCalc

extends java.lang.Object

Diese Klasse setzt ein Track-Element in Graphische Linien-Objekte um. Sie setzt ebenfalls Track-Element-Knoten in Graphische-Punkte um.

**public static void** put​(TrackElement TE,

java.awt.geom.Line2D.Double Line)

Dieses Put gibt als Key ein Trail als logisches Gleis zu dem Value eines geographischen Gleis Rail

Parameters:

TE - TrackElement - Ein logischen Gleis Trail

Line - - ein Rail - geographisches Gleise

**public static void** put​(TrackElement TE,

java.awt.geom.Point2D.Double Point)

Dieses Put gibt als Key ein Point\_RemoteOperated - ein Umschaltelement der Weiche als Track-Element zu dem Value eines BranchingSwitch – Track-Element einer Weiche

Parameters:

TE - - TrackElement - das Umschalt-Element der Weiche

Point - - BranchingSwitch - das Modell einer Weiche

**public static java.awt.geom.Line2D.Double** translateTeToGraphic​(TrackElement TE)

Transferiert ein Track-Element zu einer zeichenbaren Linie.

Parameters:

TE - - TrackElement - Ein logischen Gleis Trail

Returns:

Line2D.Double - Eine Rail die zeichenbar ist

#### Felder von Plan-Daten

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Beschreibung |
| java.util.ArrayList<BranchingSwitch> | branchingSwitchList | Liste von allen Weichen |
| tatic float | [f\_STRETCH\_X](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/model/PlanData.html#f_STRETCH_X) | Dehnungsfaktor in x-Richtung beim Zeichnen von Elementen, hier neutrales Element 1 |
| static [DefaultRepo](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/util/DefaultRepo.html)<java.lang.String,​[GeoCoordinates](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/ma/GeoCoordinates.html)> | [GeoNodeRepo](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/model/PlanData.html#GeoNodeRepo) | Ein Repository der als Schlüssel den Identifier, einen PlanPro-GeoKnoten hält und als Wert Geo-Coordinaten bereitstellt. |
| java.util.HashMap<[GradientTrailModel](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/gradient/profile/GradientTrailModel.html),​java.util.ArrayList<[GradientSegment](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/ma/GradientSegment.html)>> | [GradientMap](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/model/PlanData.html#GradientMap) | Gradientenzuordnung in HashMap key GradientTrailModel value List von GradientSegment |
| static [DefaultRepo](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/util/DefaultRepo.html)<java.lang.String,​[PlatformEdge](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/trackplan/ui/PlatformEdge.html)> | [PlatformRepo](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/model/PlanData.html#PlatformRepo) | Bahnhöfe noch nicht vollständig implementiert |
| java.util.ArrayList<[Rail](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/Rail.html)> | [railList](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/model/PlanData.html#railList) | Liste aller Gleise die man zeichnen kann |
| static [TopologyGraph](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan_pro/adapter/topology/TopologyGraph.html) | [topGraph](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/model/PlanData.html#topGraph) | Der topologische Graph speichert das SL-TMS-interene Topologische Modell |
| static int | [vmax](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/plan/elements/model/PlanData.html#vmax) | Erlaubte max. Streckengeschwindigkeit eines Gleises (Rail in km/h) |

Package de.ibw.tms.speed.profile.controller

### Class SegmentAddController

public class SegmentAddController

extends java.lang.Object

Ein Controller für Speed-Segmente. Wenn ein neues Segment zu einem Profil hinzugeordnet wird, werden UI-Komponenten benachrichtigt, sich zu aktualisieren.

#### Methoden des Segment-Add-Controllers

**public void** setModel​(SpeedSegmentViewModel model)

Model der Speed-Ansicht in der sich neue Segmente in das Wunschprofil anordnen lassen.

Parameters:

model - SpeedSegmentViewModel - View Model

**public void** setView​(AddSegmentDialog view)

Setzt den konkreten Dialog in Bezug zu dem Controller

Parameters:

view - - eine konkrete Ansicht zum Wunschprofil

**public void** submitSegment​(java.lang.String sBeginMeter, java.lang.String sEndMeter, java.lang.String stringSpeed)

Controller verarbeitet ein neues Segment und ordnet es einen Geschwindigkeitsprofil zu, das in der UI gehalten wird.

Parameters:

sBeginMeter - - Segment Beginn in Meter

sEndMeter - - Segment Ende in Meter

stringSpeed - - Speed in (km per h)

Package de.ibw.tms.speed.profile.model

### Class CartesianSpeedModel

public class CartesianSpeedModel

extends java.lang.Object

Model eines Speed Profils. Es handelt sich um einen Wrapper um die SSP-Klasse. Die SSP-Klasse beinhaltet einzelne Segmente.

#### Methoden des Segment-Add-Controllers

**public SSP** getStaticSpeedProfile()

Gibt das Speed Profil wieder.

Returns:

SSP - Statisches Geschwindigkeitsprofil

**public void** setStaticSpeedProfile​([SSP](http://localhost:63342/tms_simulator/etcs/misc/docs/SL_HTML_Dokumentatkion/SL/de/ibw/tms/ma/SSP.html) staticSpeedProfile)

Setzt ein neues Statisches Geschwindigkeitsprofil

Parameters:

staticSpeedProfile - - SSP - Profil

Package de.ibw.tms.speed.profile.model

### Class SpeedSegmentViewModel

public class SpeedSegmentViewModel

extends java.lang.Object

Ein Modell eines einzelnen Speed-Segmentes einer Anzeige. Das Modell wird für den Dialog verwendet, der angezeigt wird, wenn man in die Zeichenfläche des Geschwindigkeitsprofil-Dialoges klickt. Dort kann ein einzelnes neues Segment erstellt werden. Dieses gerade bearbeitete Segment wird durch dieses Model implementiert.

#### Konstruktor des Speed-Segment-Viewmodels

public SpeedSegmentViewModel​(SpeedSegment S)

Instanziert ein Model eines SpeedSegmentes für den Dialog eines neuen Segmentes.

Parameters:

S - SpeedSegment - das Segment dieses Models

#### Methoden von Speed-Segment-View Model

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| Chainage | getMaxChainage() | Maximaler Abstand entspricht Ende der Ma |
| Chainage | getMinChainage() | Minmaler Abstand ist die Positon des Zuges, dargestellt als Chainage mit 0 m |
| SpeedSegment | getSegment() | gibt das Segment wieder |
| void | setCtrl​(SegmentAddController ctrl) | Setzt Bezug zu dem Controller |
| void | setEndMeter​(int iResult) | End Meter des Segmentes ab Zugposition |
| void | setStartMeter​(int iResult) | Start Meter des Segmentes ab Zugposition |
| void | setMaxChainage​(Chainage maxChainage) | Setzt die maximale Ausdehnung der Segmente im Model |
| void | setMinChainage​(Chainage minChainage) | Setzt den minimalen Abstand, der sich in der UI auswählen lässt. Bisher 0 m als Positionsangabe vom Zug aus. |
| void | setSpeed​(int iResult) | Setzt die Geschwindigkeit als Eingabe in (km per h) |

Package de.ibw.tms.speed.profile.view

### Class AddSegmentDialog

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

public class AddSegmentDialog

extends javax.swing.JDialog

Der Dialog der ein einzelnes Geschwindigkeits-Segment nachbearbeitbar macht.

#### Konstruktor des Add Segment Dialoges

public AddSegmentDialog​(SpeedSegmentViewModel SegmentViewModel,

SegmentAddController Ctrl)

Dieser Konstruktor intiiert den Dialog

Parameters:

SegmentViewModel - SpeedSegmentViewModel - Model eines einzelnen Segmentes, das in diesem Dialog bearbeitet wird.

Ctrl - SegmentAddController - Controller der Aktionen ausführt

Package de.ibw.tms.speed.profile.view

### Class SpeedDialog

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

public class SpeedDialog

extends CartesianDialog

Dieser Dialog zeigt das SSP, das gesamte Geschwindigkeitsprofil einer MA an. Es besteht aus mehreren Segmenten, die im SSP enthalten sind.

#### Methode im Speed-Dialog

**public static void** displaySpeedDialog​(CartesianSpeedModel CSM,

Route RouteData)

Zeigt den SSP Dialog an

Parameters:

CSM - - ein Wrapper eines SSP

RouteData - - die momentane Route

Der Dialog zeigt das SSP aus dem Parmeter CSM an, hat aber auch Route aus RouteData.  
In der Route ist nämlich der Start und Endabstand in m definiert, wobei der Start bisher mit 0 m dem Start des Zuges entspricht.

Package de.ibw.tms.speed.profile.view

### Class SpeedPanel

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible

public class SpeedPanel

extends CartesianPanel

Panel zum SSP Dialog. Das ist der Dialog, der das gesamte Geschwindigkeitsprofil einer MA anzeigt.

#### Konstruktor des Speed-Panels

public SpeedPanel​(CartesianSpeedModel SpeedModel,

Route RouteData)

Dieser Konstruktor erstellt Komponenten des eigenen Dialogs

Parameters:

SpeedModel - CartesianSpeedModel - Model der aktuellen Ansicht, SSP-Modell

RouteData - Route - beantragte Route

#### Methoden des Speed-Panels

**public void** addSpeedSegment​(SpeedSegment InsertSegment)

Speichert ein neues Segment in das SSP ein

Parameters:

InsertSegment - SpeedSegment - das Segment das angebaut wird. Davor vorhandene Segmente enden am Start des Insert-Segmentes und beginnen am Ende des Insert-Segmentes. Vorherige Geschwindigkeiten werden auf dem des Insert-Segmentes gesetzt.

**public void** paintComponent​(java.awt.Graphics g)

Description copied from class: CartesianPanel

Zeichnet das Koordinatensystem auf dem Panel.

Overrides:

paintComponent in class CartesianPanel

Parameters:

g - Graphics - Java Zeichen Objekt

Package de.ibw.tms.trackplan

### Class EnumModel

Direct Known Subclasses:

CrossoverEnumModel

public abstract class EnumModel

extends java.lang.Object

Dieses Model verwaltet Enumerationen zum generischen Verarbeiten. Wird derzeit bei Weichen eingesetzt.

Package de.ibw.tms.trackplan

#### Sub-Class EnumModel.EnumField

Enclosing class:

EnumModel

public class EnumModel.EnumField

extends java.lang.Object

Verwaltet einen konkreten Wert eines Enums

##### Felder des Enum-Fields

**public java.lang.Enum** Item

Gerade aktiver Wert

**public java.lang.String** sName

Geade aktiver Wert als String

##### Konstruktor des Enum-Fields

public EnumField​(java.lang.Enum It,

java.lang.String sName)

Erstellt einen Value eines Enums

Parameters:

It - Enum Ein Wert eines Enums

sName - String - Enum-Wert als String

##### Methode des Enum-Fields

**public java.lang.String** toString()

Gibt gerade aktiven EnumWert als String wieder.

Overrides:

toString in class java.lang.Object

Returns:

String

#### Methoden des EnumModels

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| abstract EnumModel.EnumField[] | getEnumMappingList() | Gibt alle Werte des Enum-Felder wieder |
| EnumModel.EnumField | getFieldByDataEnum​(java.lang.Enum E) | Gibt Feld dieser Abstraktion für allgemeines Java-Enum wieder |
| EnumModel.EnumField | getSingleSelection() | Gibt gerade aktiven Enum-Wert wieder |
| void | setSingleSelection​(EnumModel.EnumField singleSelection) | Setzt neuen aktiven Wert des Enum-Models |

Package de.ibw.tms.trackplan

### Class MaCreatingFrame

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

public class MaCreatingFrame

extends javax.swing.JFrame

Fenster in dem Ma beantragt werden. Es hat ein eigene Kartenansicht.

#### Felder des Ma-Creations-Frames

public static MaCreatingFrame CurrentMaCreatingFrame

Es sollte nur eine Instanz offen sein. Sie wird hier gespeichert.

#### Konstruktor des Ma-Creations-Frames

public MaCreatingFrame​(MaRequestWrapper MovementAuthRequest, javax.swing.JFrame MainFrame)

Konstruktor diesers Ma-Erstellungsfensters

Parameters:

MovementAuthRequest - - Zu bearbeitende MA

MainFrame - - Zugeordnetes Vaterfenster

Package de.ibw.tms.trackplan

### Class TrackplanGraphicPanel

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.util.concurrent.Flow.Subscriber, javax.accessibility.Accessible

public class TrackplanGraphicPanel

extends javax.swing.JPanel

implements java.util.concurrent.Flow.Subscriber

Panel, das eine Streckenkarte beinhaltet.

#### Konstruktor des Trackplan-Graphik-Panel

public TrackplanGraphicPanel​(MaRequestWrapper Request)

Erstellt ein Panel mit der Streckenansicht

Parameters:

Request - MaRequestWrapper - zu bearbeitende MA

#### Methoden des Trackplan-Graphik-Panel

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| MaRequestWrapper | [getR](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\TrackplanGraphicPanel.html#getR())() | Gibt gerade bearbeitende MA wider. |
| Route | [getRouteModel](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\TrackplanGraphicPanel.html#getRouteModel())() | Gibt die gerade bearbeitete Route wider |
| RouteViewPort | [getRoutePort](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\TrackplanGraphicPanel.html#getRoutePort())() | Gibt Vermittlung von Route zum Fenster wider. |
| void | [onComplete](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\TrackplanGraphicPanel.html#onComplete())() | Zeichnet sich neu nach Erhalt eines Befehls. |
| void | [onError](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\TrackplanGraphicPanel.html#onError(java.lang.Throwable))​(java.lang.Throwable t) | Behandelt Fehler bei Empfang von Nachricht |
| void | [onNext](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\TrackplanGraphicPanel.html#onNext(java.lang.Object))​(java.lang.Object item) | Bereitet neuen Empfang vor, nach Erhalt dieser Nachricht „item“. |
| void | [onSubscribe](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\TrackplanGraphicPanel.html#onSubscribe(java.util.concurrent.Flow.Subscription))​(java.util.concurrent.Flow.Subscription subscription) | Meldet Sich als Empfänger von Neuzeichen-Befehlen an |
| void | [paintComponent](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\TrackplanGraphicPanel.html#paintComponent(java.awt.Graphics))​(java.awt.Graphics g) | Zeichnet die Kartenansicht in dieses Panel |
| void | [setRoutePort](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\TrackplanGraphicPanel.html#setRoutePort(de.ibw.tms.trackplan.ui.RouteViewPort))​([RouteViewPort](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\ui\RouteViewPort.html) routePort) | Gibt die Route einer MA an, die gerade geplant werden soll. Die Klasse RouteViewPort stellt eine Vermittlung dar zur Route-Klasse dar. |

Package de.ibw.tms.trackplan.controller

### Class CrossoverController

All Implemented Interfaces:

IController<CrossoverMainModel>,java.lang.AutoCloseable, java.util.concurrent.Flow.Publisher<CrossoverMainModel>

public class CrossoverController

extends java.util.concurrent.SubmissionPublisher<CrossoverMainModel>

implements IController<CrossoverMainModel>

Verwaltet den Status einer Weiche

#### Konstruktor des Crossover-Controller

public CrossoverController​(CrossoverMainModel Model,

CrossoverEnumModel CrossStatModel,

BranchingSwitch C)

Dieser Konstruktor hat ein Weiche als Model, den Status und die geographische Weiche

Parameters:

Model - CrossoverModel - Modell der Weiche

CrossStatModel - - CrossoverEnumModel - Status der Weiche

C - BranchingSwitch - Position der Weiche

#### Methoden des Crossover-Controller

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| void | [addSubscriber](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\CrossoverController.html#addSubscriber(java.util.concurrent.Flow.Subscriber))​(java.util.concurrent.Flow.Subscriber toSubscribe) | Schreibt sich eine Komponente ein, werden Information bei Statuswechsel gesendet. |
| java.util.List<java.util.concurrent.Flow.Subscriber> | [getSubscriberList](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\CrossoverController.html#getSubscriberList())() | gibt alle eingeschriebenen Komponenten wider |
| void | [publish](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\CrossoverController.html#publish())() | Benachrichtigt alle eingeschriebenen Komponenten |

Package de.ibw.tms.trackplan.controller

### Class MoveController

All Implemented Interfaces:

IController, java.lang.AutoCloseable, java.util.concurrent.Flow.Publisher

public class MoveController

extends java.util.concurrent.SubmissionPublisher

implements IController

Verwaltet Interaktion mit Nutzer, wenn dieser die Zeichnung verschiebt.

#### Methoden des Move-Controller

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| static MoveController | [getInstance](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\MoveController.html#getInstance())() | Singleton dieses Controllers |
| java.util.List<java.util.concurrent.Flow.Subscriber> | [getSubscriberList](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\MoveController.html#getSubscriberList())() | List aller Benachrichtigungs-Empfänger |
| void | [publish](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\MoveController.html#publish())() | Move wird an eingeschriebene Komponenten benachrichtigt |
| void | [setTranslation](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\MoveController.html#setTranslation(int,int))​(int dx, int dy) | Setzt eine Koordinatenverschiebung ab.  Parameters:  dx - - wie weit wird X geschoben  dy - - wie weit wird Y geschoben |

Package de.ibw.tms.trackplan.controller

### Class RouteController

All Implemented Interfaces:

IController<Route>, java.lang.AutoCloseable, java.util.concurrent.Flow.Publisher<Route>

public class RouteController

extends java.util.concurrent.SubmissionPublisher<Route>

implements IController<Route>

Verwaltet beantragte Routen.

#### Konstruktor des Route-Controller

public RouteController​(RouteViewPort routePort)

Instanziiert diesen Kontroller

Parameters:

routePort - - Vermittlung zur Route

#### Methoden des Route-Controller

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| Route | getRouteData() | Die verwaltete Route |
| java.util.List<java.util.concurrent.Flow.Subscriber> | getSubscriberList() | Liste aller Nachrichten-Empfänger |
| void | publish() | Benachrichtig alle Komponenten die sich registriert haben, dass eine Route geändert wurde. |
| void | setRouteData​(Route R) | setzt die vom Controller verwaltete Route |

Package de.ibw.tms.trackplan.controller

### Class TrackController

All Implemented Interfaces:

IController<java.lang.String>, java.lang.AutoCloseable, java.util.concurrent.Flow.Publisher<java.lang.String>

public class TrackController

extends java.util.concurrent.SubmissionPublisher<java.lang.String>

implements IController<java.lang.String>

Verwaltet MAs. MAs dieses Controllers werden in einem Track-Dialog angelegt. Dieser hat wie das Hauptfenster eine Karte.

#### Felder des Track-Controllers

**public static java.awt.Point** ClickPoint

Speichert letzten Punkt, den der Nutzer angeklickt hat.

#### Methoden des Track-Controllers

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| void | [deleteMaAction](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\TrackController.html#deleteMaAction())() | Entfernt eine entworfene nicht gesendete MA. |
| void | [editMaAction](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\TrackController.html#editMaAction())() | Vorhandene MA soll bearbeitet werden. |
| static [TrackController](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\TrackController.html) | [getInstance](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\TrackController.html#getInstance(de.ibw.tms.trackplan.TrackplanGraphicPanel))​([TrackplanGraphicPanel](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\TrackplanGraphicPanel.html) TrackPanel) | Holt den Kontroller der zum angegebenen Fenster gehört.  Parameters:  TrackPanel - - Das Fenster zu dem der Controller angefragt wird  Returns:  TrackController - Verwaltung der MAs, dieses Fensters |
| java.util.List<java.util.concurrent.Flow.Subscriber> | [getSubscriberList](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\TrackController.html#getSubscriberList())() | Liste aller zu benachrichtigten Komponenten. |
| void | [handleMousePress](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\TrackController.html#handleMousePress(java.awt.Point,boolean))​(java.awt.Point point, boolean isMainWindow) | Verwaltet Klicks auf der Zeichenebene. |
| void | [publish](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\TrackController.html#publish())() | MA Veränderung wird weitergegeben |
| void | [requestMaAction](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\TrackController.html#requestMaAction())() | Beantragt, dass Nutzer eine neue Ma definieren will. |
| void | [setRouteViewPort](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\TrackController.html#setRouteViewPort(de.ibw.tms.trackplan.ui.RouteViewPort))​([RouteViewPort](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\ui\RouteViewPort.html) routePort) | Setzt die Vermittlung zu Routen |

Package de.ibw.tms.trackplan.controller

### Class ZoomController

All Implemented Interfaces:

IController, java.lang.AutoCloseable, java.util.concurrent.Flow.Publisher

public class ZoomController

extends java.util.concurrent.SubmissionPublisher

implements IController

Verwaltet das Zoomen in Karten

#### Methoden des Zoom-Controller

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| void | [applyZoom](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\ZoomController.html#applyZoom())() | Wendet Zoom an, der in der GUI eingestellt wurde. |
| void | [changeX](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\ZoomController.html#changeX(java.lang.String))​(java.lang.String sXZoom) | Setzt den Zoom in x Richtung auf sXZoom |
| void | [changeY](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\ZoomController.html#changeY(java.lang.String))​(java.lang.String sYZoom) | Setzt Wert des Zooms in y-Richtung |
| void | [focusTrain](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\ZoomController.html#focusTrain(de.ibw.tms.train.model.TrainModel))​([TrainModel](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\train\model\TrainModel.html) trainModel) | Setzt einen Zug in Fokus |
| static [ZoomController](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\ZoomController.html) | [getInstance](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\ZoomController.html#getInstance())() | Singleton holt den Controller |
| java.util.List<java.util.concurrent.Flow.Subscriber> | [getSubscriberList](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\ZoomController.html#getSubscriberList())() | Liste aller Zoom-Nachrichten - Empfänger |
| void | [publish](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\ZoomController.html#publish())() | Benachrichtig alle registrierten Komponenten, das es einen neuen Zoom Faktor gibt |
| void | [zoomIn](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\ZoomController.html#zoomIn())() | Verdoppeln des Zoom-Faktors |
| void | [zoomOut](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\controller\ZoomController.html#zoomOut())() | Halbieren des Zoom-Faktors |

Package de.ibw.tms.trackplan.ui

### Class LinearLocationWaypointSetWindow

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

public class LinearLocationWaypointSetWindow

extends javax.swing.JDialog

Verwaltet Distanzfeld des Routenendes auf dem letzten Gleis einer topologischen Kante.

#### Konstruktor des Linear-Location-Waypoint-Set-Windows

public LinearLocationWaypointSetWindow​(TrackElement TE,

javax.swing.JFrame Frame,

java.awt.Point P,

RouteController RC,

TrainModel startingPointTrain)

Erstellt den Dialog mit Schiebebalken. Er definiert den Abstand zum Zug.

Parameters:

TE - - Letzte Element der Route der MA

Frame - - Zu welchen Frame wird der Dialog gesetzt.

P - - Position dieses Dialoges

RC -- RouteController - Verwaltung falls neue Route angelegt wird

startingPointTrain - TrainModel - Zug zu dieser MA

#### Methoden des Linear-Location-Waypoint-Set-Windows

public Route getRouteModel()

Holt Route zu diesem Dialog

Returns:

Route

Package de.ibw.tms.trackplan.ui

### Class MainGraphicPanel

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.util.concurrent.Flow.Subscriber, javax.accessibility.Accessible

public class MainGraphicPanel

extends javax.swing.JPanel

implements java.util.concurrent.Flow.Subscriber

Das Panel des Hauptfensters

#### Methoden des Main-Graphic-Panels

public static void paintTrains​(java.awt.Graphics2D g2d)

Zeichnet Züge in das Fenster

Parameters:

g2d - Graphics2D - Zeichenutil

public void paintComponent​(java.awt.Graphics g)

Zeichnet die Karte im Hauptfenster

Overrides:

paintComponent in class javax.swing.JComponent

Parameters:

g - - Zeichenutil

public static void paintCrossroad​(TranslationModel t\_Model,

ZoomModel zoom, java.awt.Graphics2D g2d)

Zeichnet alle Weichen

Parameters:

t\_Model - TranslationModel - Nachberechnung der Position der Weichen.

Das Model gibt an wo der Anzeigeausschnitt sich befindet.

zoom – ZoomModel – Wird auch zur Nachberechnung der Weichenposition verwendet

g2d - Graphics2D -- Zeichenutil

public static void paintBalises​(TranslationModel t\_Model, ZoomModel zoom,

java.awt.Graphics2D g2d)

Zeichnet Balisen

Parameters:

t\_Model - TranslationModel - Nachberechnung der Position der Balisen.

zoom – ZoomModel - Nachberechnung der Position der Balisen.

g2d - Graphics2D - Zeichenutil

public static [GeoCoordinates](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\ma\GeoCoordinates.html) getGeoCoordinate​(java.lang.String TopKanteId,

boolean b\_fromA,

double distanceA1)

Berechnet die GeoCoordinate zu einer Topologischen Kante mit Bezug zu einem Knoten mit Distanz.

Parameters:

TopKanteId - String - Knoten Id PlanPro

b\_fromA - boolean - ist von A gemessen worden

distanceA1 - double - Abstand zum Referenzknoten

Returns:

GeoCoordinates - Geographischer Punkt

public void onSubscribe​(java.util.concurrent.Flow.Subscription subscription)

Schreibt sich ein, als Impuls, wann neugezeichnet werden soll

Specified by:

onSubscribe in interface java.util.concurrent.Flow.Subscriber

Parameters:

subscription – Die Einzeichnung fürs Neuzeichnen

public void onNext​(java.lang.Object planData)

Andere Komponente ruft neuzeichnen hervor

Specified by:

onNext in interface java.util.concurrent.Flow.Subscriber

Parameters:

planData - - unused

public void onError​(java.lang.Throwable throwable)

Fehler beim Neuzeichnen, genauer beim Erhalt der Nachricht zum Neuzeichnen

Specified by:

onError in interface java.util.concurrent.Flow.Subscriber

Parameters:

throwable - - Fehler

public void onComplete()

definiert, was in Zusatz getan wird, wenn neuzeichnen ankommt.

Specified by:

onComplete in interface java.util.concurrent.Flow.Subscriber

Package de.ibw.tms.trackplan.ui

### Class RouteComponent

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.util.concurrent.Flow.Subscriber<Route>, javax.accessibility.Accessible

public class RouteComponent

extends javax.swing.JPanel

implements java.util.concurrent.Flow.Subscriber<Route>

Dieses Panel stellt ein Kontextmenu dar. Die Aktionen dieses Menus betreffen das Setzten der Waypoints einer MA Route.

#### Felder der Route-Component

|  |  |  |
| --- | --- | --- |
| Datentyp | Felder | Beschreibung |
| static [CartesianSpeedModel](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\speed\profile\model\CartesianSpeedModel.html) | [CSM](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\ui\RouteComponent.html#CSM) | Kapselt das angeforderte Geschwindigkeitsprofil der gesamten Strecke. |
| static java.util.Stack<[TrackElement](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\ma\physical\TrackElement.html)> | [lastTrackElements](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\ui\RouteComponent.html#lastTrackElements) | Stack zum Speichern der Reihenfolge in der Waypoints als TrackElement angelegt werden. |
| static [SvL](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\ma\SvL.html) | [svl](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\ui\RouteComponent.html#svl) | Subervised Location, wird einem Zusatzdialog gesetzt. |

#### Kontruktoren der Route-Component

public RouteComponent​(TrackElement TrackEl, RouteController RC)

Erstellt das Anzeige-Feld innerhalb eines MA-Anlege-Panels. Wenn dort in die Zeichenebene geklickt wird, wird für jedes naheliegende Shape eine solche Routen-Komponente innerhalb eines neuen JFrame angelegt.

Parameters:

TrackEl - TrackElement - Ein Element, für das eine RouteComponent angelegt wird.

RC - RouteComponent - Führt Aktionen zum Verändern von Routen aus.

public RouteComponent​(java.lang.String desc, TrackElement trackReference,

RouteController routeCntrl)

Erstellt das Panel für ein Auswahlmenu von TrackElement (meinst Knoten)

Parameters:

desc - - Name Menus-Knopfes in diesem Kontextmenu-Panel.

trackReference - TrackElement - Element das dieses Menu betrifft

routeCntrl - RouteController - Stellt Aktionen zum Ändern der Route in dieser Komponente zur Verfügung

#### Methoden der Route-Component

public static double calcTrackLengthUntilLastWayoint​(TrainModel TrModel)

Berechnet die Streckenlänge in Metern des letzten Streckenabschnitts. Wird benötigt um den Nutzer eine Auswahl zu geben, wo der Zug am letzten Streckenabschnitt enden soll. Das ist dann das Ende der MA. Die

Es wird die Entfernung des letzten Topologie-Knotens bis zum Haltepunkt auf der letzten Kante der MA als double zurückgegeben.

Parameters:

TrModel - TrainModel - Zug Model des Zuges, dem die MA betrifft

Returns:

double - Strecke in Meter

public void onSubscribe​(java.util.concurrent.Flow.Subscription subscription)

Schreibt sich ein über Routenänderungen informiert zu werden, um sich neu zeichnen zu können

Specified by:

onSubscribe in interface java.util.concurrent.Flow.Subscriber<[Route](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\ma\Route.html)>

Parameters:

subscription - Flow.Subscription - Einschreibung

public void onNext​([Route](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\ma\Route.html) item)

Routenveränderung trifft ein.

Specified by:

onNext in interface java.util.concurrent.Flow.Subscriber<Route>

Parameters:

item - Route

public void onError​(java.lang.Throwable throwable)

Behandelt Fehler

Specified by:

onError in interface java.util.concurrent.Flow.Subscriber<Route>

Parameters:

throwable - - Fehler

public void onComplete()

Bei Routenänderungen wird neu gezeichnet.

Specified by:

onComplete in interface java.util.concurrent.Flow.Subscriber<Route>

Package de.ibw.tms.trackplan.ui

### Class RouteViewPort

All Implemented Interfaces:

java.util.concurrent.Flow.Subscriber<Route>

public class RouteViewPort

extends java.lang.Object

implements java.util.concurrent.Flow.Subscriber<Route>

Stellt Beziehung zu genau einer Route eines Zuges zu dem Streckenfenster einer MA her.

#### Konstruktor des Route-View-Ports

public RouteViewPort​(Route RouteModel,

TrackplanGraphicPanel trackPanel)

Erstellt den Bezug von Route zum MA-Streckenfenster

Parameters:

RouteModel – Route -

trackPanel - in diesem Streckenfenster

#### Methoden des Route-View-Ports

public Route getRouteModel()

Gibt aktuelle Route im Streckenfester wider.

Returns:

Route

public void setRouteModel​(Route routeModel)

Setzt die Route einer Ma in dem Streckenfenster

Parameters:

routeModel - Route - R

public void onSubscribe​(java.util.concurrent.Flow.Subscription subscription)

hört auf Nachrichten von Routenänderungen

Specified by:

onSubscribe in interface java.util.concurrent.Flow.Subscriber<Route>

Parameters:

subscription -

public void onNext​(Route item)

Routenänderung umsetzen.

Specified by:

onNext in interface java.util.concurrent.Flow.Subscriber<Route>

Parameters:

item - Route

public void onError​(java.lang.Throwable throwable)

Fehler bei eingehender Routennachricht

Specified by:

onError in interface java.util.concurrent.Flow.Subscriber<Route>

Parameters:

throwable - Throwable Fehler

public void onComplete()

Nach verarbeiten der Nachricht wird neugezeichnet.

Specified by:

onComplete in interface java.util.concurrent.Flow.Subscriber<Route>

public void paintRoute​(java.awt.Graphics2D g2d)

Zeichnet Routen in das Ma-Streckenfenster ein-

Parameters:

g2d - Graphics2D - Zeichenutil

Package [de.ibw.tms.trackplan.ui](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\ui\package-summary.html)

### Class SingleEnumSelectorComponent<T extends EnumModel>

Type Parameters:

T - EnumModel - Art des verwalteten Enums.

All Implemented Interfaces:

java.awt.event.ActionListener, java.awt.image.ImageObserver, java.awt.ItemSelectable, java.awt.MenuContainer, java.io.Serializable, java.util.EventListener, javax.accessibility.Accessible, javax.swing.event.ListDataListener

public class SingleEnumSelectorComponent<T extends EnumModel>

extends javax.swing.JComboBox

Allgemeine Komponente zum Ändern von Enums. Wird derzeit nur bei den Weichen eingesetzt. Verwaltet Enum-Werte in einer Combobox.

#### Konstruktor des Single-Enum-Selektors

public SingleEnumSelectorComponent​(T SelectionModel, IController C)

Initiiert Komponente, wenn ein Fenster den Weichenstatus setzt

Parameters:

SelectionModel - - Modell des Enums

C - - Gibt gesetzte Werte an alle registrierten Komponenten weiter.

#### Methode des Single-Enum-Selektors

public void update​(java.lang.Enum Selected)

Setzt den Status des Enums, weil das Enum eventuell an einer anderen Stelle gesetzt wurde.

Parameters:

Selected - Enum - gesetzter Wert des Enums

Package de.ibw.tms.trackplan.ui

### Class TrackPanel

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible

public class TrackPanel

extends javax.swing.JPanel

Panel, das innerhalb eines Rechts-Click Dialogs verwendet wird. Betrifft genau ein Element in der Kartenansicht, das sich im Klickbereich befand.

#### Konstruktor des Track-Panel

public TrackPanel​(Iinteractable Model, RouteViewPort RoutePort, boolean isMainWindow)

Erstellt das Panel zu einem Element, das sich in der Kartenansicht in der Nähe des Rechtsclick befand.

Parameters:

Model - Iinteractable - Model des Elments das in diesem Panel verarbeitet wird.

RoutePort - RouteViewPort – Port, umin dieser Komponente Routeninformation der Ma anzupassen

isMainWindow - boolean - ist dieses Panel im Hauptfenster des TMS angesteuert worden

Package de.ibw.tms.trackplan.ui

### Class TrackWindow

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

public class TrackWindow

extends javax.swing.JFrame

Neuer Frame der alle Elementbearbeitungs-Optionen bei Rechtsklick in der Streckenansicht wiedergibt.

#### Konstruktor des Track-Windows

public TrackWindow​(java.util.List panels, java.awt.Point P)

Erstellt ein Rechtsklick-Fenster.

Parameters:

panels - List eine Liste von Panels je Elementes im Klickbereich.

P - Point Punkt der im Streckenfester geklickt wurde.

#### Methode von dem Track-Window

public static void closeAllTrackWindows()

Entfernt alle Rechtsclck-Fenster.

Package de.ibw.tms.trackplan.ui

### Class WaypointDecorator

All Implemented Interfaces:

IPaintable, IWaypoint, java.io.Serializable

public class WaypointDecorator

extends Waypoint

implements IWaypoint, IPaintable

Ein Wrapper von Waypoints. Start und Endpunkt sind nicht als Waypoints modelliert worden. Diese Klasse macht auch für Start- oder End-Knoten - Waypoint-Funktionen verfügbar.

#### Konstruktor des Waypoint-Dekorators

public WaypointDecorator​(TrackElement startOrEndElement,

TrackElementStatus TeStatus,

int xPixel,

int yPixel)

Erweitert Start- oder End-Waypoint

Parameters:

startOrEndElement - TrackElement – Element, das Wrapper haben soll

TeStatus - TrackElementStatus - Status des angefragten Elementes

xPixel - int - xPosition des Bildes in der Streckenansicht

yPixel - int - yPosition des Bildes in der Streckenansicht

#### Methode des Waypoint-Dekorators

public void paintImage​(java.awt.Graphics2D g2d)

Zeichnet Bild des Dekorateurs mit Beschriftung.

Specified by:

paintImage in interface IPaintable

Parameters:

g2d - Graphics2D Zeichenutil.

Package de.ibw.tms.trackplan.ui

### Class WaypointEnd

All Implemented Interfaces:

IPaintable, IWaypoint, java.io.Serializable

public class WaypointEnd

extends SpotLocation

implements IWaypoint, IPaintable

Der End-Waypoint einer Route einer Ma-Beantragung

#### Konstruktor von Waypoint-End

public WaypointEnd​(Chainage C,

TrackElement endElement,

SectionOfLine Section,

int xPixel,

int yPixel)

Generiert End-Waypoint

Parameters:

C - - unused

endElement - - letztes Element der Route

Section - - unused

xPixel - int x-Zeichenpositon - nicht reale Position in der Streckenansicht.

yPixel - int y-Zeichenpositon - nicht reale Position in der Streckenansicht

#### Methoden des Waypoint-Ends

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| Chainage | getChainage() | unused |
| TrackElement | getTrackElement() | Gibt Element von Waypoint wider |
| void | paintImage​(java.awt.Graphics2D g2d) | Zeichnet Bild des End-Waypoints. |

Package de.ibw.tms.trackplan.ui

### Class WaypointStart

All Implemented Interfaces:

IPaintable, IWaypoint, java.io.Serializable

public class WaypointStart

extends SpotLocation

implements IWaypoint, IPaintable

Der Start-Waypoint einer Route einer Ma-Beantragung. Wird durch Zug-Position ersetzt und nicht mehr benutzt.

#### Konstruktor Waypoint-Start

public WaypointStart​(Chainage C,

TrackElement startElement,

SectionOfLine Section,

int xPixel,

int yPixel)

Generiert Start-Waypoint

Parameters:

C - - unused

startElement - - erstes Element der Route

Section - - unused

xPixel - int x-Zeichenposition - nicht reale Position in der Streckenansicht.

yPixel - int y-Zeichenposition - nicht reale Position in der Streckenansicht.

#### Methoden des Waypoint-Start

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| Chainage | getChainage() | unused |
| TrackElement | getTrackElement() | Gibt Track-Element wider |
| void | paintImage​(java.awt.Graphics2D g2d) | Zeichnet Startpunkt |

Package de.ibw.tms.trackplan.ui

### Class ZoomFrame

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

public class ZoomFrame

extends javax.swing.JFrame

Fenster zum Einstellen des Zoom-Faktors

#### Methoden des Zoom-Frames

public static [ZoomFrame](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\trackplan\ui\ZoomFrame.html) getZoomFrame()

Singelton - Gibt Zoom Fenster wider

Returns:

ZoomFrame - Das Zoom-Fenster

public static java.util.concurrent.Flow.Subscriber getPositionSubscriber()

Horcht auf Positionswechsel von Zügen, damit das Auswahlfeld der Zugreferenz aktuell bleibt.

Returns:

Subscriber - ein Listener auf Positionswechsel

Package de.ibw.tms.trackplan.ui

### Class ZoomPanel

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible

public class ZoomPanel

extends javax.swing.JPanel

Panel zum Einstellen des Zoom-Faktors.

#### Konstruktor des Zoom-Panels

public ZoomPanel​(javax.swing.JFrame parentFrame)

Erstellt die Kompontenten des Zoom Fensters

Parameters:

parentFrame - JFrame - das Zoom Fenster

#### Methode des Zoom-Panels

public void setTrainBoxEntries​(java.util.Collection<TrainModel> trainModels)

Setzt die registrierten Züge.

Parameters:

trainModels - Collection alle Zugmodelle, die registriert sind

Package de.ibw.tms.trackplan.ui

### Enum LinksRechts

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<LinksRechts>, java.lang.constant.Constable

public enum LinksRechts

extends java.lang.Enum<LinksRechts>

Verwaltet Lage zum Gleis in Bahnhöfen

Package de.ibw.tms.trackplan.viewmodel

### Class TranslationModel

public class TranslationModel

extends java.lang.Object

Dieses Model verwaltet die Position des angezeigten Streckenausschnitts innerhalb der Streckenansichten.

#### Sub-Class TrackplanEnvironment

Enclosing class:

TranslationModel

public static class TranslationModel.TrackplanEnvironment

extends java.lang.Object

Diese Klasse verwaltet die Start-Positionen der unterschiedlichen Karten. Hier ist der Start-Auschnitt der Karte gemeint.

##### Felder des Environments

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Erklärung |
| static TranslationModel.TrackplanEnvironment | CurrentEnvironment | Gerade benutztes Strecken-Setting |
| java.lang.String | resourceLocation | Ort der Plan-Pro-Datei in den Ressourcen des Java-Packages |
| ZoomModel | Zoom | Aktiver Zoom-Faktor als Model |

##### Konstruktor des Environments

public TrackplanEnvironment​(TranslationModel translation,

ZoomModel zoom,

java.lang.String resourceLocation)

Erstellt ein neues Setting mit Startpostion, Startzoom und Ort der Plan-Pro-Datei als Name in den Java-Resourcen

Parameters:

translation - TranslationModel - Startposition der Kartenansicht

zoom - ZoomModel - Startzoom der Kartenansicht

resourceLocation - String - Name der Plan-Pro-Datei, als Resourcen-Name

#### Methoden des Translation Models

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| double | getdMoveX() | X-Position der betrachteten Kartenansicht |
| double | getdMoveY() | Y-Position der betrachteten Kartenansicht |
| static TranslationModel | getInstance() | Gibt Aktuelle Karten-Position als Model wider |
| void | setdMoveX​(double dMoveX) | Setzt neue X-Position des betrachteten Kartenausschnittes. |
| void | setdMoveY​(double dMoveY) | Setzt neue Y-Position des betrachteten Kartenausschnittes. |

Package de.ibw.tms.trackplan.viewmodel

### Class ZoomModel

public class ZoomModel

extends java.lang.Object

Dieses Model verwaltet den Zoom innerhalb der Streckenansicht.

#### Methoden des Zoom-Models

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| double | getdZoomX() | Gibt den Zoomfaktor in x-Richtung wider |
| double | getdZoomY() | Gibt den Zoomfaktor in y-Richtung wider |
| static ZoomModel | getInstance() | Singleton gibt aktuelles Zoom Model wider |
| java.lang.String | getsInfo() | Status Nachricht, falls Nutzerfehler entstanden |
| void | setdZoomX​(double dZoomX) | setzt den Zoom in x-Richtung |
| void | setdZoomY​(double dZoomY) | setzt den Zoom in Y-Richtung |
| void | setsInfo​(java.lang.String sInfo) | Setzt Nachricht bei Fehler |

Package de.ibw.tms.train.controller

### Class TrainController

All Implemented Interfaces:

IController, java.lang.AutoCloseable, java.util.concurrent.Flow.Publisher

public class TrainController

extends java.util.concurrent.SubmissionPublisher

implements IController

Controller, der die Metadaten zu dem Zug verwaltet. Das kann im MA-Beantragungsfenster unterhalb der Karte editiert werden.

#### Konstruktor des Train-Controllers

public TrainController​(TrainModel Tm,

SingleTrainSubPanel Tsp)

Instanziert Controller zu einem Zugmodell. Der Controller kann die Daten zum Zug ändern.

Parameters:

Tm – TrainModel – Beschreibt die Daten zu einem Zug

Tsp – SingleTrainSubPanel – Das Panel, das UI-Elemente beinhaltet, die den Zug bearbeitbar machen.

#### Methoden des Train-Controllers

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| void | changeCategory​(java.lang.String sCategory) | Setzt Art des Zuges auf den angegebenen Wert |
| void | changeLabel​(java.lang.String sLabel) | Ändert Name eines Zuges |
| void | changeLength​(java.lang.String sLength) | Setzt Länge des Zuges auf angegebenen Wert in Meter |
| void | changeSpeed​(java.lang.String sSpeed) | Setzt die maximale Geschwindigkeit des Zuges |
| void | changeTraincolor​(java.awt.Color RepresentedColor) | Ändert Farbe eines Zuges. |
| LinkingProfileAdapter | extractLinkingProfile​(Route R) | TODO Linking in der MA |
| java.util.List<java.util.concurrent.Flow.Subscriber> | getSubscriberList() | Definiert welche Komponenten benachrichtigt werden. |
| void | publish() | Gibt Auskunft, dass ein Zug editiert wurde. |
| void | setCSM​(CartesianSpeedModel CSM) | Setzt das Static Speed Profile des Zuges |

public void requestMovementAuthority​(MaRequestWrapper requestWrapper,

Route R,

java.lang.String sRbcId,

java.lang.String sTmsId,

java.util.UUID uuid,

boolean bIsShunting)

throws java.io.IOException

Diese Methode schickt eine Movement-Authority an die SmartLogic. Die SmartLogic prüft die Daten.

Parameters:

requestWrapper - MaRequestWrapper - Die Anfrage einer MA an die SL

R - Route - die angeforderte Route

sRbcId - String - Ziel RBC Id

sTmsId - String - Eigene TMS Id

uuid - - UUID - Kommunikations Id

bIsShunting - boolean - Entscheidet ob Shunting gefahren werden darf, wenn das Ende der MA vom Zug erreicht wurde

Throws:

java.io.IOException - - Fehler beim Senden an die SmartLogic

Package de.ibw.tms.train.model

### Class TrainDistance

public class TrainDistance

extends java.lang.Object

Abstand des Zuges von einem Knoten einer Topologischen Kante. Der Zug soll sich auf dieser Kante befinden.

#### Konstruktor von Train-Distance

public TrainDistance​(boolean isA,

double d1,

double d2)

Generiert einen Abstand von einem Topologischen Knoten, der referenzierten Kante.

Parameters:

isA - boolean - Gibt an ob A der referenzierte Knoten ist oder B

d1 - double Abstand zum Heck des Zuges

d2 - double Abstand zur Zugnase

#### Methoden der Train-Distance

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| double | getdDistance1() | Abstand vom Heck des Zuges zum Referenzknoten |
| double | getdDistance2() | Abstand von der Zugnase zum Referenzknoten |
| boolean | isIsfromA() | Wird Abstand von Knoten a dann (true) gemessen oder von Knoten b dann (false) |

Package de.ibw.tms.train.model

### Class TrainModel

public class TrainModel

extends java.lang.Object

Modell eines Zuges, das bei Position Reports neu angelegt wird.

#### Felder des Trainmodels

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Erklärung |
| java.lang.String | category | Kategorie des Zuges |
| int | iSpeedMax | Maximale Geschwindigkeit des Zuges |
| int | iTrainId | Nid-Engine-Id als Integer |
| java.lang.String | label | Label als String, normalerweise die Nid Engine Id als String |
| double | length | Ausdehnung des Zuges |
| java.awt.Color | RepresentedColor | Farbe des Zuges auf der Zeichenebene |
| static [ThreadedRepo](file:///C:\Workspace\Javadoc\SL\de\ibw\util\ThreadedRepo.html)<java.lang.Integer,​[TrainModel](file:///C:\Workspace\Javadoc\SL\de\ibw\tms\train\model\TrainModel.html)> | TrainRepo | Verwaltet alle Züge als Repository, Keys sind die Nid-Engine-ID, Values das jeweilige Modell. |
| static java.util.ArrayList<java.awt.Color> | usedColorList | Verwaltet benutzte Zugfarben als Liste |
| static java.util.ArrayList<java.lang.String> | usedLabelList | Verwaltet benutzte Zugbezeichner als Liste |

Methoden des Trainmodels

Hier werde die Getter-Methoden beschreiben, weil das hier Zusammenhänge besser eindringlich macht.

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| double | getdDistanceToNodeRunningTo() | Gibt Entfernung zum Toplogischen Knoten wider, auf dem sich der Zug hinbewegt. |
| static TrainModel | getDefaultModel() | Gibt ein Standard-Zug wider |
| TrainDistance | getDistanceRefPointOfEdge() | Gibt Abstand zum Topologischen Referenzknoten auf der Topologischen Zugkante wider |
| TopologyGraph.Edge | getEdgeTrainStandsOn() | Gibt die Toplogische Kante, auf die der Zug sich befindet wider. |
| java.lang.Integer | getNid\_lrbg() | Gibt die Nid-Id des Zuges wider. |
| TopologyGraph.Node | getNodeTrainRunningTo() | Gibt den topologischen Knoten auf den sich der Zug hinbewegt wider. |
| java.lang.String | getsLastKnownRBC() | Gibt die Id des Letzen bekannten RBCs dieses Zug-Models wider |
| int | getTrainId() | Gib Nid-Engine-ID wider |
| java.awt.geom.Line2D.Double | getTrainUiLine() | Gibt zu zeichnende Linde des Zuges wider |
| java.lang.String | toString() | Zug Nid-Engine-Id |

Package de.ibw.tms.train.ui

### Class SingleTrainSubPanel

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, java.util.concurrent.Flow.Subscriber<TrainModel>, javax.accessibility.Accessible

public class SingleTrainSubPanel

extends javax.swing.JPanel

implements java.util.concurrent.Flow.Subscriber<TrainModel>

Das Panel befindet sich unterhalb der Karte beim Erstellen einer MA.

#### Felder des SingleTrainSubPanels

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Erklärung |
| javax.swing.JFrame | Parent | Frame der dieses Panel beinhaltet |
| static SingleTrainSubPanel | TrainPanel | Die aktuelle Instanz des Panels |
| static float | xFactor | Positons-Skalierung von Bilder sollte 1.0f bleiben |

#### Konstruktor des SingleTrainSubPanels

public SingleTrainSubPanel​(TrackplanGraphicPanel TGP,

javax.swing.JFrame parentFrame)

Instanziiert dieses Panel unterhalb der Streckenansicht bei den MA-Bantragungs-UIs.

Parameters:

TGP - TrackplanGraphicPanel - das Panel behinhaltet die Karte und dieses Panel

parentFrame - JFrame - Frame der übergeordnet ist.

#### Methoden des SingleTrainSubPanel

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| void | onComplete() | Wenn Zug-Daten ankommen wird das Hauptfenster mit allen Subkomponenten erneuert. |
| void | onError​(java.lang.Throwable t) | Fehler bei Verarbeitung ankommender Zug-Daten |
| void | onNext​(TrainModel item) | Neue Daten des Zuges |
| void | onSubscribe​(java.util.concurrent.Flow.Subscription subscription) | Schreibt diese Klasse als horchend auf, wenn Zug-Daten ankommen. |

Package de.ibw.tms.train.ui

### Class TrainPanel

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible

public class TrainPanel

extends javax.swing.JPanel

Das Panel beinhaltet die Kartenansicht einer neuen MA Erstellung. Und auch das Panel darunter, dass Zug-Daten verwaltet.

#### Konstruktor des Train-Panel

public TrainPanel​(javax.swing.JFrame frame,

TrackplanGraphicPanel trackPanel,

MaRequestWrapper Request)

Erstellt vom Hauptfenster zum Editieren einer MA ein neues Fenster.

Parameters:

frame - JFrame - Der Frame des Hauptfensters

trackPanel - TrackplanGraphicPanel - Die Zeichenfläche oder Karte.

Request - MaRequestWrapper - Der Request der durch diese UI vom Nutzer erstellt wird.

Package de.ibw.util

### Class DefaultRepo<K,​V>

Direct Known Subclasses:

ThreadedRepo

public class DefaultRepo<K,​V>

extends java.lang.Object

Ein Repository gebaut aus einer Hashmap

#### Methoden des DefaultRepo

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| boolean | containsKey​(K Key) | Sucht ob Key schon in diesem Repository existiert. |
| java.util.Collection<V> | getAll() | Gibt alle Values dieses Repositorien ohne Keys wider |
| V | getModel​(K key) | Holt zum key den Wert |
| void | removeKey​(K key) | Entfernt Value mit dem Key, beides wird entfernt |
| java.util.List<V> | sortValues() | Sortiert diese Repository nach deren Keys und gibt eine Sortierte Liste der Values wider. |
| void | update​(K key, V mapValue) | Gibt zu einem Key einen Wert in die HashMap zu. |

Package de.ibw.util

### Class DoubleCoord

All Implemented Interfaces:

ICoord<java.lang.Double>

public class DoubleCoord

extends java.lang.Object

implements [ICoord](file:///C:\Workspace\Javadoc\SL\de\ibw\util\ICoord.html)<java.lang.Double>

Diese Klasse verwaltet Positionen als Double. Es wird eine X und ein Y-Koordinate verwaltet

Package de.ibw.util

### Class StatusMessageAppender

All Implemented Interfaces:

org.apache.log4j.Appender, org.apache.log4j.spi.OptionHandler

public class StatusMessageAppender

extends org.apache.log4j.AppenderSkeleton

Diese Klasse gibt das Logging in ein Textfeld.

#### Methoden des Status—Message-Appenders

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Erklärung |
| protected void | append​(org.apache.log4j.spi.LoggingEvent event) | Schreibt ein Log in das Textfeld |
| void | close() | Entfernt den Appender. |
| boolean | requiresLayout() | Signalisiert Appender ohne Layout. |

Package de.ibw.util

### Class ThreadedRepo<K,​V>

public class ThreadedRepo<K,​V>

extends DefaultRepo<K,​V>

Ein Threadsicheres DefaultRepo, auch ein Repository.

#### Methoden des ThreadedRepo

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| java.util.Collection<V> | getAll() | Gibt alle Values dieses Repositories ohne Keys wider |
| V | getModel​(K key) | Holt zum key den Wert |
| void | removeKey​(K key) | Entfernt Value mit dem Key beides wird entfernt |
| void | update​(K key, V mapValue) | Gibt zu einem Key einen Wert in die HashMap zu. |

Package de.ibw.util

### Class UtilFunction

public class UtilFunction

extends java.lang.Object

Uitilities Allgemeiner Art

#### Methoden als Util-Functions

public static int formatStringToInt​(java.lang.String sInputSpeed)

throws java.lang.NumberFormatException

Trennt Einheit von Wert. Gibt den Wert als int wider.

Parameters:

sInputSpeed - String - Wert und Einheit als Zeichenkette

Returns:

int

Throws:

java.lang.NumberFormatException - - Wenn Wert sich nicht als Integer dargestellt werden kann.

public static [ICoord](file:///C:\Workspace\Javadoc\SL\de\ibw\util\ICoord.html)<java.lang.Double> calcTargetGeoByStartPoint​(ICoord<java.lang.Double> CalcTarget, double dA, GeoCoordinates geo\_A, GeoCoordinates geo\_B)

Gegeben sind zwei Coordinaten geo\_A und Geo\_B. Es wird eine Linie über diese Koordinaten virtuell realisiert. Diese Linie wird nicht gezeichnet, deswegen sei sie virtuell bezeichnet. In der Linie auf den beiden Coordinate wird der Punkt widergegeben der den Abstand dA von Coordinate A hat.

A->da->Ziel->db->B

Parameters:

CalcTarget - ICoord Die ZielCoordinate, das Ergebnis

dA - - Abstand zu Coordinate A

geo\_A - - GeoCoordinates A

geo\_B - GeoCoordinates B

Returns:

ICoord - Ergebnis

Package ebd

### Class ConfigHandler

Diese Klasse verwaltet die Parameter der sl\_config-Datei im Ordner Konfiguration. Es wurde Lars Schultze-Falk Code neu verwendet.

#### Felder des Config-Handlers

|  |  |  |
| --- | --- | --- |
| Datentyp | Feld | Erklärung |
| int | entityID | Id des TMS oder der SL |
| java.lang.String | entityName | Bezeichner des TMS oder SL |
| java.lang.String | ipToGUIServer4SL | IP des Gui-Server für Lognachrichten in der SL |
| java.lang.String | ipToGUIServer4TMS | IP des Gui-Server für Lognachrichten im TMS |
| java.lang.String | portOfGUIServer4SL | Port des SL-Gui-Server |
| java.lang.String | portOfGUIServer4TMS | Port des TMS-Gui-Server |

#### Methoden des Config-Handlers

|  |  |  |
| --- | --- | --- |
| Datentyp | Methode | Beschreibung |
| static ConfigHandler | getInstance() | Singleton des Config-Handlers |
| void | loadDefault() | Lädt die Parameter |
| void | saveCurrent() | Speichert die Parameter |

## Die Generischen Klassen aus Planpro

Die generischen Klassen aus Plan Pro wurden über die Schma.xsd Datei des PlanPro-Formats generiert. Es wurden nur wenige Klassen geändert, um die Balisen zu unterstützen, aber auch allgemein um Fehler zu vermeiden. Um diese Klassen zu verstehen eignen sich die Beschreibungen des Plan-Pro-Glossar sehr gut.