

Enoclean

1.0

Generated by Doxygen 1.8.12

Contents

1	Namespace Index	1
1.1	Packages	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	Package ch	9
5.2	Package ch.bfh	9
5.3	Package ch.bfh.gr33nopo55um	9
5.4	Package ch.bfh.gr33nopo55um.enocean	9
5.5	Package ch.bfh.gr33nopo55um.enocean.application	10
5.6	Package ch.bfh.gr33nopo55um.enocean.helper	10
5.7	Package ch.bfh.gr33nopo55um.enocean.persistence	10
5.8	Package ch.bfh.gr33nopo55um.enocean.telegram	10
5.9	Package ch.bfh.gr33nopo55um.enocean.test	10

6	Class Documentation	11
6.1	All	11
6.1.1	Detailed Description	11
6.2	An	12
6.2.1	Detailed Description	12
6.3	Application	12
6.3.1	Detailed Description	13
6.3.2	Member Function Documentation	13
6.3.2.1	main()	13
6.4	Checksum	13
6.4.1	Detailed Description	13
6.5	Class	14
6.5.1	Detailed Description	14
6.6	Class	14
6.6.1	Detailed Description	14
6.7	CommonCommand	15
6.7.1	Detailed Description	17
6.7.2	Member Function Documentation	17
6.7.2.1	decodeTelegramData()	17
6.7.2.2	decodeTelegramHeader()	18
6.7.2.3	dumpData()	18
6.7.2.4	encodeTelegram()	19
6.7.2.5	encodeTelegramData()	19
6.7.2.6	getCcCode()	19
6.7.2.7	getCcData()	19
6.7.2.8	setCcCode()	19
6.7.2.9	setCcData()	20
6.7.2.10	toString()	20
6.7.3	Member Data Documentation	20
6.7.3.1	ccCode	20

6.7.3.2	ccData	20
6.7.3.3	id	21
6.8	CRC	21
6.8.1	Detailed Description	21
6.8.2	Member Function Documentation	21
6.8.2.1	calc()	21
6.8.3	Member Data Documentation	22
6.8.3.1	crc8Table	22
6.9	DB	22
6.9.1	Detailed Description	22
6.10	diagnosis	23
6.10.1	Detailed Description	23
6.11	Dummy	24
6.11.1	Detailed Description	26
6.11.2	Member Function Documentation	26
6.11.2.1	decodeTelegramData()	26
6.11.2.2	dumpData()	26
6.11.2.3	encodeTelegram()	26
6.11.2.4	encodeTelegramData()	27
6.11.2.5	toString()	27
6.11.3	Member Data Documentation	27
6.11.3.1	id	27
6.12	EncodeDecode	28
6.12.1	Detailed Description	29
6.12.2	Member Function Documentation	29
6.12.2.1	decodeTelegram()	29
6.12.2.2	decodeTelegramData()	29
6.12.2.3	decodeTelegramHeader()	30
6.12.2.4	dump()	30
6.12.2.5	dumpData()	30

6.12.2.6	dumpHeader()	31
6.12.2.7	encodeTelegram()	31
6.12.2.8	encodeTelegramData()	31
6.12.2.9	encodeTelegramHeader()	32
6.12.2.10	persist()	32
6.13	EnocaelListener	32
6.13.1	Detailed Description	33
6.13.2	Member Function Documentation	33
6.13.2.1	logger()	33
6.13.3	Member Data Documentation	33
6.13.3.1	logger	33
6.14	ERP1	33
6.14.1	Detailed Description	34
6.15	ERP2	34
6.15.1	Detailed Description	34
6.16	Event	35
6.16.1	Detailed Description	37
6.16.2	Member Function Documentation	37
6.16.2.1	decodeTelegramData()	37
6.16.2.2	dumpData()	38
6.16.2.3	encodeTelegram()	38
6.16.2.4	encodeTelegramData()	39
6.16.2.5	getEventCode()	39
6.16.2.6	getEventData()	39
6.16.2.7	getId()	39
6.16.2.8	setEventCode()	39
6.16.2.9	setEventData()	40
6.16.2.10	setId()	40
6.16.2.11	toString()	40
6.16.3	Member Data Documentation	40

6.16.3.1	eventCode	40
6.16.3.2	EventData	40
6.16.3.3	id	41
6.17	Logic	41
6.17.1	Detailed Description	41
6.18	Main	41
6.18.1	Detailed Description	42
6.19	Physical	42
6.19.1	Detailed Description	42
6.20	Radio	42
6.20.1	Detailed Description	43
6.21	Radio802	44
6.21.1	Detailed Description	46
6.21.2	Member Function Documentation	46
6.21.2.1	decodeTelegramData()	46
6.21.2.2	dumpData()	47
6.21.2.3	encodeTelegram()	48
6.21.2.4	encodeTelegramData()	48
6.21.2.5	getDbm()	48
6.21.2.6	getDestinationId()	48
6.21.2.7	getId()	48
6.21.2.8	getMessageData()	49
6.21.2.9	getMessageRorg()	49
6.21.2.10	getSourceId()	49
6.21.2.11	setDbm()	49
6.21.2.12	setDestinationId()	49
6.21.2.13	setId()	50
6.21.2.14	setMessageData()	50
6.21.2.15	setMessageRorg()	50
6.21.2.16	setSourceId()	51

6.21.2.17 toString()	51
6.21.3 Member Data Documentation	51
6.21.3.1 dbm	51
6.21.3.2 destinationId	51
6.21.3.3 id	52
6.21.3.4 messageData	52
6.21.3.5 messageRorg	52
6.21.3.6 sourceId	52
6.22 RadioERP1	53
6.22.1 Detailed Description	55
6.22.2 Member Function Documentation	55
6.22.2.1 decodeTelegramData()	55
6.22.2.2 dumpData()	56
6.22.2.3 encodeTelegram()	57
6.22.2.4 encodeTelegramData()	57
6.22.2.5 getDbm()	57
6.22.2.6 getDestinationID()	57
6.22.2.7 getId()	57
6.22.2.8 getSecurityLevel()	58
6.22.2.9 getSubTelNum()	58
6.22.2.10 setDbm()	58
6.22.2.11 setDestinationID()	58
6.22.2.12 setId()	59
6.22.2.13 setSecurityLevel()	59
6.22.2.14 setSubTelNum()	59
6.22.2.15 toString()	60
6.22.3 Member Data Documentation	60
6.22.3.1 dbm	60
6.22.3.2 destinationID	60
6.22.3.3 id	60

6.22.3.4	securityLevel	60
6.22.3.5	subTelNum	61
6.23	RadioERP2	62
6.23.1	Detailed Description	64
6.23.2	Member Function Documentation	64
6.23.2.1	decodeTelegramData()	64
6.23.2.2	dumpData()	65
6.23.2.3	encodeTelegram()	65
6.23.2.4	encodeTelegramData()	66
6.23.2.5	getDbm()	66
6.23.2.6	getId()	66
6.23.2.7	getRawData()	66
6.23.2.8	getSubTelNum()	66
6.23.2.9	setDbm()	67
6.23.2.10	setId()	67
6.23.2.11	setRawData()	67
6.23.2.12	setSubTelNum()	68
6.23.2.13	toString()	68
6.23.3	Member Data Documentation	68
6.23.3.1	dbm	68
6.23.3.2	id	68
6.23.3.3	rawData	69
6.23.3.4	subTelNum	69
6.24	RadioMessage	70
6.24.1	Detailed Description	72
6.24.2	Member Function Documentation	72
6.24.2.1	decodeTelegramData()	72
6.24.2.2	dumpData()	73
6.24.2.3	encodeTelegram()	73
6.24.2.4	encodeTelegramData()	74

6.24.2.5	getDbm()	74
6.24.2.6	getDestinationID()	74
6.24.2.7	getId()	74
6.24.2.8	getSourceID()	74
6.24.2.9	setDbm()	74
6.24.2.10	setDestinationID()	75
6.24.2.11	setId()	75
6.24.2.12	setSourceID()	75
6.24.2.13	toString()	76
6.24.3	Member Data Documentation	76
6.24.3.1	dbm	76
6.24.3.2	destinationID	76
6.24.3.3	id	76
6.24.3.4	sourceID	76
6.25	RadioSubTel	77
6.25.1	Detailed Description	79
6.25.2	Member Function Documentation	79
6.25.2.1	decodeTelegramData()	79
6.25.2.2	dumpData()	80
6.25.2.3	encodeTelegram()	81
6.25.2.4	encodeTelegramData()	81
6.25.2.5	getDbm()	81
6.25.2.6	getDbmSubTel()	81
6.25.2.7	getDestinationID()	82
6.25.2.8	getSecurityLevel()	82
6.25.2.9	getStatusSubTel()	82
6.25.2.10	getSubTelNum()	82
6.25.2.11	getTickSubTel()	82
6.25.2.12	getTimeStamp()	82
6.25.2.13	setDbm()	82

6.25.2.14	getDbmSubTel()	83
6.25.2.15	setDestinationID()	83
6.25.2.16	setSecurityLevel()	83
6.25.2.17	setStatusSubTel()	84
6.25.2.18	setSubTelNum()	84
6.25.2.19	setTickSubTel()	84
6.25.2.20	setTimeStamp()	85
6.25.2.21	toString()	85
6.25.3	Member Data Documentation	85
6.25.3.1	dbm	85
6.25.3.2	dbmSubTel	85
6.25.3.3	destinationID	86
6.25.3.4	id	86
6.25.3.5	securityLevel	86
6.25.3.6	statusSubTel	86
6.25.3.7	subTelNum	86
6.25.3.8	tickSubTel	86
6.25.3.9	timeStamp	86
6.26	ReadConfig	87
6.26.1	Detailed Description	87
6.26.2	Member Function Documentation	87
6.26.2.1	readPropertyValue()	87
6.26.3	Member Data Documentation	88
6.26.3.1	inputStream	88
6.27	remote	88
6.27.1	Detailed Description	88
6.28	RemoteManCommand	89
6.28.1	Detailed Description	91
6.28.2	Member Function Documentation	91
6.28.2.1	decodeTelegramData()	91

6.28.2.2	dumpData()	92
6.28.2.3	encodeTelegram()	92
6.28.2.4	encodeTelegramData()	93
6.28.2.5	getDbm()	93
6.28.2.6	getDestinationID()	93
6.28.2.7	getSendDelay()	93
6.28.2.8	getSourceID()	93
6.28.2.9	setDbm()	93
6.28.2.10	setDestinationID()	94
6.28.2.11	setSendDelay()	94
6.28.2.12	setSourceID()	94
6.28.2.13	toString()	95
6.28.3	Member Data Documentation	95
6.28.3.1	dbm	95
6.28.3.2	destinationID	95
6.28.3.3	id	95
6.28.3.4	sendDelay	95
6.28.3.5	sourceID	96
6.29	Response	97
6.29.1	Detailed Description	99
6.29.2	Member Function Documentation	99
6.29.2.1	decodeTelegramData()	99
6.29.2.2	dumpData()	100
6.29.2.3	encodeTelegram()	100
6.29.2.4	encodeTelegramData()	101
6.29.2.5	getReturnCode()	101
6.29.2.6	setReturnCode()	101
6.29.2.7	toString()	101
6.29.3	Member Data Documentation	102
6.29.3.1	id	102

6.29.3.2	returnCode	102
6.30	SeedDB	102
6.30.1	Detailed Description	103
6.30.2	Constructor & Destructor Documentation	103
6.30.2.1	SeedDB()	103
6.30.3	Member Function Documentation	103
6.30.3.1	getInstance()	103
6.30.3.2	insertCoreData()	104
6.30.4	Member Data Documentation	104
6.30.4.1	instance	104
6.31	Sender	105
6.31.1	Detailed Description	105
6.31.2	Member Function Documentation	105
6.31.2.1	main()	105
6.32	Serveral	106
6.32.1	Detailed Description	106
6.33	SMART_ACK_COMMAND	106
6.33.1	Detailed Description	106
6.34	SmartAckCommand	107
6.34.1	Detailed Description	109
6.34.2	Member Function Documentation	109
6.34.2.1	decodeTelegramData()	109
6.34.2.2	dumpData()	110
6.34.2.3	encodeTelegram()	110
6.34.2.4	encodeTelegramData()	111
6.34.2.5	getSmartAckCommandData()	111
6.34.2.6	getSmartAckCommandNr()	111
6.34.2.7	setSmartAckCommandData()	111
6.34.2.8	setSmartAckCommandNr()	111
6.34.2.9	toString()	112

6.34.3	Member Data Documentation	112
6.34.3.1	id	112
6.34.3.2	smartAckCommandData	112
6.34.3.3	smartAckCommandNr	112
6.35	Store	113
6.35.1	Detailed Description	113
6.35.2	Member Function Documentation	113
6.35.2.1	createQuery()	113
6.35.2.2	getEntityManager()	114
6.35.2.3	getInstance()	114
6.35.3	Member Data Documentation	115
6.35.3.1	entityManager	115
6.35.3.2	instance	115
6.35.3.3	jpaStoreId	115
6.36	TelegramDBFactory	115
6.36.1	Detailed Description	116
6.36.2	Member Function Documentation	116
6.36.2.1	telegramRouter()	116
6.36.3	Member Data Documentation	117
6.36.3.1	logger	117
6.37	TelegramHeader	117
6.37.1	Detailed Description	119
6.37.2	Member Function Documentation	119
6.37.2.1	decodeTelegram()	119
6.37.2.2	decodeTelegramHeader()	120
6.37.2.3	dump()	121
6.37.2.4	dumpHeader()	122
6.37.2.5	encodeTelegram()	122
6.37.2.6	encodeTelegramHeader()	123
6.37.2.7	getCrcData()	123

6.37.2.8	getCrcHeader()	124
6.37.2.9	getCreationDate()	124
6.37.2.10	getData()	124
6.37.2.11	getDataLength()	124
6.37.2.12	getFullTelegram()	124
6.37.2.13	getOptionalLenght()	125
6.37.2.14	getSyncByte()	125
6.37.2.15	persist()	125
6.37.2.16	setCrcData()	125
6.37.2.17	setCrcHeader()	126
6.37.2.18	setCreationDate()	127
6.37.2.19	setData()	127
6.37.2.20	setDataLength()	128
6.37.2.21	setFullTelegram()	128
6.37.2.22	setOptionalLenght()	129
6.37.2.23	setSyncByte()	129
6.37.3	Member Data Documentation	130
6.37.3.1	crcData	130
6.37.3.2	crcHeader	130
6.37.3.3	creationDate	130
6.37.3.4	data	130
6.37.3.5	dataLength	130
6.37.3.6	fullTelegram	130
6.37.3.7	optionalLenght	130
6.37.3.8	syncByte	131
6.38	Tester	131
6.38.1	Detailed Description	131
6.38.2	Member Function Documentation	132
6.38.2.1	main()	132
6.39	This	132

6.39.1 Detailed Description	132
6.40 Time	133
6.40.1 Detailed Description	133
6.41 TimeLogger	133
6.41.1 Detailed Description	134
6.41.2 Member Function Documentation	134
6.41.2.1 endLogTime()	134
6.41.2.2 getOperationName()	135
6.41.2.3 getStartTime()	135
6.41.2.4 getStopTime()	136
6.41.2.5 setOperationName()	136
6.41.2.6 setStartTime()	136
6.41.2.7 setStopTime()	136
6.41.2.8 startLogTime()	137
6.41.3 Member Data Documentation	137
6.41.3.1 logger	137
6.41.3.2 operationName	137
6.41.3.3 startTime	137
6.41.3.4 stopTime	137
6.42 Used	138
6.42.1 Detailed Description	138
6.43 When	138
6.43.1 Detailed Description	138

7	File Documentation	139
7.1	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/application/Application.java File Reference	139
7.2	Application.java	139
7.3	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/application/Enoccea↔ Listener.java File Reference	139
7.4	EnocceaListener.java	140
7.5	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/helper/CRC.java File Refer- ence	140
7.6	CRC.java	141
7.7	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/helper/ReadConfig.java File Reference	142
7.8	ReadConfig.java	142
7.9	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/helper/TimeLogger.java File Reference	142
7.10	TimeLogger.java	143
7.11	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/persistence/SeedDB.java File Reference	143
7.12	SeedDB.java	144
7.13	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/persistence/Store.java File Reference	144
7.14	Store.java	144
7.15	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/persistence/TelegramDB↔ Factory.java File Reference	145
7.16	TelegramDBFactory.java	145
7.17	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/CommonCommand.java File Reference	146
7.18	CommonCommand.java	146
7.19	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/Dummy.java File Reference	147
7.20	Dummy.java	148
7.21	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/EncodeDecode.java File Reference	148
7.22	EncodeDecode.java	148
7.23	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/Event.java File Reference	149

7.24	Event.java	149
7.25	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/Radio802.java File Reference	150
7.26	Radio802.java	150
7.27	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/RadioERP1.java File Reference	152
7.28	RadioERP1.java	152
7.29	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/RadioERP2.java File Reference	153
7.30	RadioERP2.java	154
7.31	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/RadioMessage.java File Reference	155
7.32	RadioMessage.java	155
7.33	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/RadioSubTel.java File Reference	156
7.34	RadioSubTel.java	156
7.35	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/RemoteMan↔ Command.java File Reference	158
7.36	RemoteManCommand.java	158
7.37	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/Response.java File Reference	160
7.38	Response.java	160
7.39	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/SmartAck↔ Command.java File Reference	161
7.40	SmartAckCommand.java	161
7.41	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/Telegram↔ Header.java File Reference	162
7.42	TelegramHeader.java	162
7.43	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/test/ch/bfh/gr33nopo55um/enoccean/test/↔ Sender.java File Reference	164
7.44	Sender.java	164
7.45	/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/test/ch/bfh/gr33nopo55um/enoccean/test/↔ Tester.java File Reference	165
7.46	Tester.java	165

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

ch	9
ch.bfh	9
ch.bfh.gr33nopo55um	9
ch.bfh.gr33nopo55um.enocean	9
ch.bfh.gr33nopo55um.enocean.application	10
ch.bfh.gr33nopo55um.enocean.helper	10
ch.bfh.gr33nopo55um.enocean.persistence	10
ch.bfh.gr33nopo55um.enocean.telegram	10
ch.bfh.gr33nopo55um.enocean.test	10

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

All	11
An	12
Application	12
Checksum	13
Class	14
Class	14
CRC	21
DB	22
diagnosis	23
EncodeDecode	28
TelegramHeader	117
CommonCommand	15
Dummy	24
Event	35
Radio802	44
RadioERP1	53
RadioERP2	62
RadioMessage	70
RadioSubTel	77
RemoteManCommand	89
Response	97
SmartAckCommand	107
EnocleaListener	32
ERP1	33
ERP2	34
Logic	41
Main	41
Physical	42
Radio	42
ReadConfig	87
remote	88
SeedDB	102
Sender	105
Serveral	106
SMART_ACK_COMMAND	106

Store	113
TelegramDBFactory	115
Tester	131
This	132
Time	133
TimeLogger	133
Used	138
When	138

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

All	11
An	12
Application	12
Checksum	13
Class	14
Class	14
CommonCommand	15
CRC	21
DB	22
diagnosis	23
Dummy	24
EncodeDecode	28
EnocleaListener	32
ERP1	33
ERP2	34
Event	35
Logic	41
Main	41
Physical	42
Radio	42
Radio802	44
RadioERP1	53
RadioERP2	62
RadioMessage	70
RadioSubTel	77
ReadConfig	87
remote	88
RemoteManCommand	89
Response	97
SeedDB	102
Sender	105
Serveral	106
SMART_ACK_COMMAND	106
SmartAckCommand	107
Store	113

TelegramDBFactory	115
TelegramHeader	117
Tester	131
This	132
Time	133
TimeLogger	133
Used	138
When	138

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/application/ Application.java	139
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/application/ EnocleaListener.java	139
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/ CRC.java	140
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/ ReadConfig.java	142
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/ TimeLogger.java	142
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/persistence/ SeedDB.java	143
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/persistence/ Store.java	144
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/persistence/ TelegramDB	↔
Factory.java	145
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ CommonCommand	↔
java	146
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ Dummy.java	147
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ EncodeDecode.java	148
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ Event.java	149
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ Radio802.java	150
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ RadioERP1.java	152
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ RadioERP2.java	153
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ RadioMessage.java	155
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ RadioSubTel.java	156
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ RemoteManCommand	↔
java	158
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ Response.java	160
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ SmartAckCommand	↔
java	161
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ TelegramHeader.java	162
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/test/ch/bfh/gr33nopo55um/enOcean/test/ Sender	↔
java	164
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/test/ch/bfh/gr33nopo55um/enOcean/test/ Tester	↔
java	165

Chapter 5

Namespace Documentation

5.1 Package ch

Packages

- package **bfh**

5.2 Package ch.bfh

Packages

- package **gr33nopo55um**

5.3 Package ch.bfh.gr33nopo55um

Packages

- package **enoccean**

5.4 Package ch.bfh.gr33nopo55um.enoccean

Packages

- package **application**
- package **helper**
- package **persistence**
- package **telegram**
- package **test**

5.5 Package ch.bfh.gr33nopo55um.enocean.application

Classes

- class **Application**
- class **EnoceaListener**

5.6 Package ch.bfh.gr33nopo55um.enocean.helper

Classes

- class **CRC**
- class **ReadConfig**
- class **TimeLogger**

5.7 Package ch.bfh.gr33nopo55um.enocean.persistence

Classes

- class **SeedDB**
- class **Store**
- class **TelegramDBFactory**

5.8 Package ch.bfh.gr33nopo55um.enocean.telegram

Classes

- class **CommonCommand**
- class **Dummy**
- interface **EncodeDecode**
- class **Event**
- class **Radio802**
- class **RadioERP1**
- class **RadioERP2**
- class **RadioMessage**
- class **RadioSubTel**
- class **RemoteManCommand**
- class **Response**
- class **SmartAckCommand**
- class **TelegramHeader**

5.9 Package ch.bfh.gr33nopo55um.enocean.test

Classes

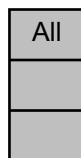
- class **Sender**
- class **Tester**

Chapter 6

Class Documentation

6.1 All

Collaboration diagram for All:



6.1.1 Detailed Description

Author

silas & louis cannot parsed into other tables.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**Dummy.java**

6.2 An

Collaboration diagram for An:



6.2.1 Detailed Description

Author

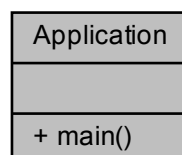
silas & louis primarily a confirmation for processes and procedures, incl. specific data content. For further informations check EnOceanSerialProtocol v3

The documentation for this class was generated from the following file:

- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/Event.java`

6.3 Application

Collaboration diagram for Application:



Static Public Member Functions

- static void **main** (String args[]) throws Exception

6.3.1 Detailed Description

Definition at line 7 of file **Application.java**.

6.3.2 Member Function Documentation

6.3.2.1 main()

```
static void main (  
    String args[] ) throws Exception [static]
```

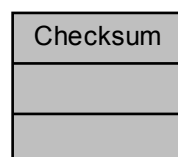
Definition at line 9 of file **Application.java**.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/application/**Application.java**

6.4 Checksum

Collaboration diagram for Checksum:



6.4.1 Detailed Description

Author

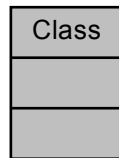
silas

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/**CRC.java**

6.5 Class

Collaboration diagram for Class:



6.5.1 Detailed Description

Author

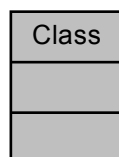
silas the config file

The documentation for this class was generated from the following file:

- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/ReadConfig.java`

6.6 Class

Collaboration diagram for Class:



6.6.1 Detailed Description

Author

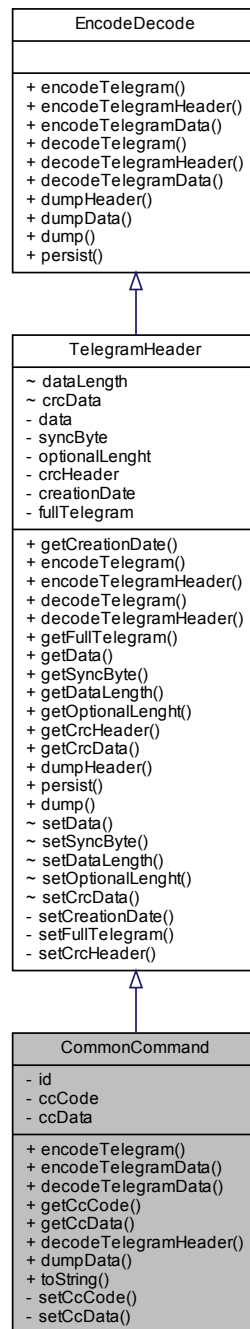
silas the config file

The documentation for this class was generated from the following file:

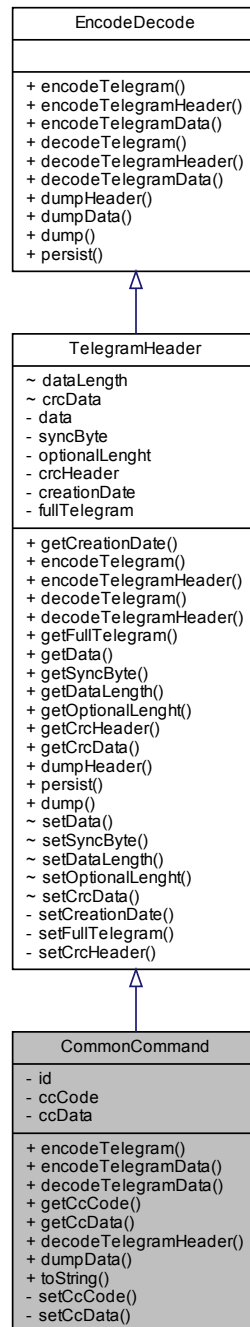
- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/ReadConfig.java`

6.7 CommonCommand

Inheritance diagram for CommonCommand:



Collaboration diagram for CommonCommand:



Public Member Functions

- String **encodeTelegram** ()
encodeTelegram provides an example telegram hex for this packet type.
- String **encodeTelegramData** ()
encodeTelegramData provides an example data hex for this packet type.
- void **decodeTelegramData** (String hexTelegram)

decodeTelegramData splits telegram data in parts

- int **getCcCode** ()
- int **getCcData** ()
- void **decodeTelegramHeader** (String hexTelegram)
- void **dumpData** ()

dumpData Print Data, useful for logs and testing EnOcean Telegram

- String **toString** ()

Private Member Functions

- void **setCcCode** (int ccCode)
- void **setCcData** (int ccData)

Private Attributes

- Long **id**
- int **ccCode**
- int **ccData**

Additional Inherited Members

6.7.1 Detailed Description

Definition at line **14** of file **CommonCommand.java**.

6.7.2 Member Function Documentation

6.7.2.1 decodeTelegramData()

```
void decodeTelegramData (
    String hexTelegram )
```

decodeTelegramData splits telegram data in parts

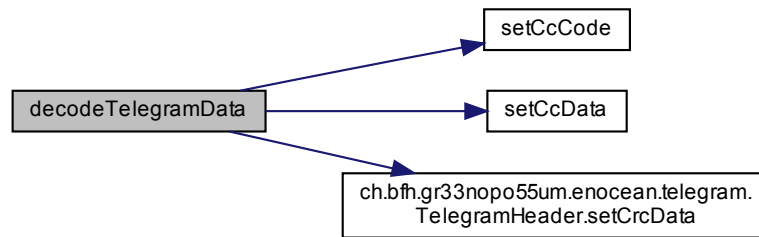
Parameters

<i>hexTelegram</i>	EnOcean Telegram
--------------------	------------------

Implements **EncodeDecode** (p. ??).

Definition at line **40** of file **CommonCommand.java**.

Here is the call graph for this function:



6.7.2.2 decodeTelegramHeader()

```
void decodeTelegramHeader (
    String hexTelegram )
```

`decodeHeaderTelegram` splits telegram in parts

Parameters

<i>hexTelegram</i>	EnOcean Telegram
--------------------	------------------

Implements **EncodeDecode** (p. ??).

Definition at line 71 of file **CommonCommand.java**.

6.7.2.3 dumpData()

```
void dumpData ( )
```

`dumpData` Print Data, useful for logs and testing EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line 79 of file **CommonCommand.java**.

Here is the call graph for this function:



6.7.2.4 encodeTelegram()

```
String encodeTelegram ( )
```

encodeTelegram provides an example telegram hex for this packet type.

Returns

String hexTelegram EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line 26 of file **CommonCommand.java**.

6.7.2.5 encodeTelegramData()

```
String encodeTelegramData ( )
```

encodeTelegramData provides an example data hex for this packet type.

Returns

String hexTelegramData EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line 35 of file **CommonCommand.java**.

6.7.2.6 getCcCode()

```
int getCcCode ( )
```

Definition at line 49 of file **CommonCommand.java**.

6.7.2.7 getCcData()

```
int getCcData ( )
```

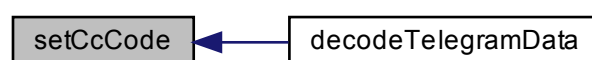
Definition at line 57 of file **CommonCommand.java**.

6.7.2.8 setCcCode()

```
void setCcCode (
    int ccCode ) [private]
```

Definition at line 53 of file **CommonCommand.java**.

Here is the caller graph for this function:

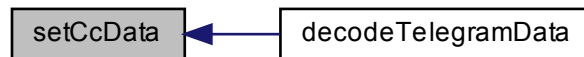


6.7.2.9 setCcData()

```
void setCcData (
    int ccData ) [private]
```

Definition at line **61** of file **CommonCommand.java**.

Here is the caller graph for this function:



6.7.2.10 toString()

```
String toString ( )
```

Definition at line **86** of file **CommonCommand.java**.

Here is the caller graph for this function:



6.7.3 Member Data Documentation

6.7.3.1 ccCode

```
int ccCode [private]
```

Definition at line **22** of file **CommonCommand.java**.

6.7.3.2 ccData

```
int ccData [private]
```

Definition at line **23** of file **CommonCommand.java**.

6.7.3.3 id

```
Long id [private]
```

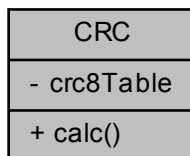
Definition at line **19** of file **CommonCommand.java**.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**CommonCommand.java**

6.8 CRC

Collaboration diagram for CRC:



Static Public Member Functions

- static byte **calc** (byte[] data)

Static Private Attributes

- static byte [] **crc8Table**

6.8.1 Detailed Description

Definition at line **8** of file **CRC.java**.

6.8.2 Member Function Documentation

6.8.2.1 calc()

```
static byte calc (
    byte [] data ) [static]
```

Definition at line **44** of file **CRC.java**.

6.8.3 Member Data Documentation

6.8.3.1 crc8Table

```
byte [] crc8Table [static], [private]
```

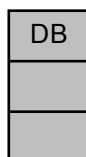
Definition at line 10 of file **CRC.java**.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/**CRC.java**

6.9 DB

Collaboration diagram for DB:



6.9.1 Detailed Description

initial generating

Author

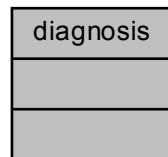
silas

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/persistence/**SeedDB.java**

6.10 diagnosis

Collaboration diagram for diagnosis:



6.10.1 Detailed Description

Author

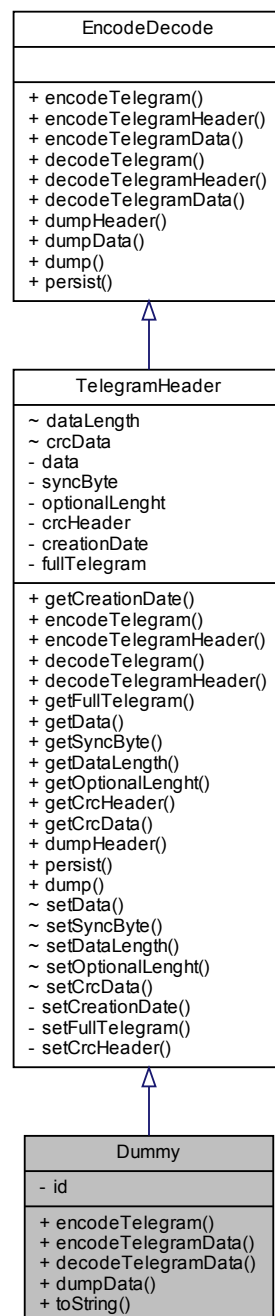
silas & louis . The packet design corresponds to the type RADIO_ERP1. The content of the OPTIONAL_DATA is altered slightly, for further informations check EnocceanSerialProtocol v3

The documentation for this class was generated from the following file:

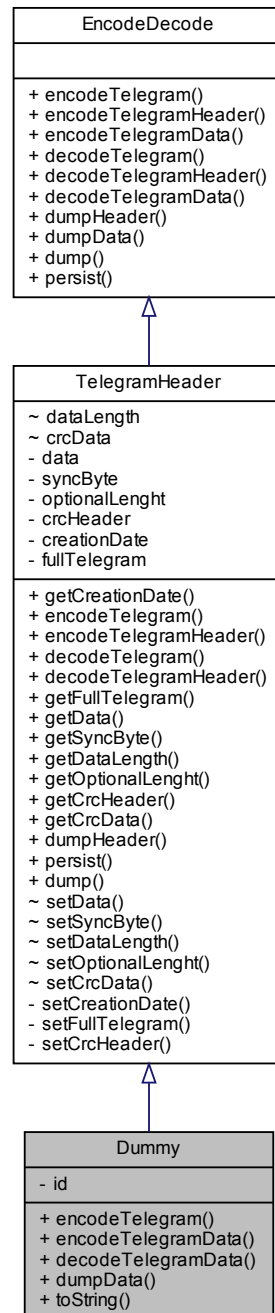
- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/**RadioSubTel.java**

6.11 Dummy

Inheritance diagram for Dummy:



Collaboration diagram for Dummy:



Public Member Functions

- String **encodeTelegram** ()
encodeTelegram provides an example telegram hex for this packet type.
- String **encodeTelegramData** ()
- void **decodeTelegramData** (String hexTelegram)
- void **dumpData** ()

usually a dummy package does not have any data

- String **toString** ()

Private Attributes

- Long **id**

Additional Inherited Members

6.11.1 Detailed Description

Definition at line **13** of file **Dummy.java**.

6.11.2 Member Function Documentation

6.11.2.1 decodeTelegramData()

```
void decodeTelegramData (
    String hexTelegram )
```

Parameters

<i>hexTelegram</i>	EnOcean Telegram
--------------------	------------------

Implements **EncodeDecode** (p. ??).

Definition at line **40** of file **Dummy.java**.

6.11.2.2 dumpData()

```
void dumpData ( )
```

usually a dummy package does not have any data

Implements **EncodeDecode** (p. ??).

Definition at line **48** of file **Dummy.java**.

6.11.2.3 encodeTelegram()

```
String encodeTelegram ( )
```

encodeTelegram provides an example telegram hex for this packet type.

Returns

String hexTelegram EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line **21** of file **Dummy.java**.

6.11.2.4 encodeTelegramData()

```
String encodeTelegramData ( )
```

encodeTelegramData provides an example data hex for this packet type.

Returns

String hexTelegramData EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line **31** of file **Dummy.java**.

6.11.2.5 toString()

```
String toString ( )
```

Returns

PacketType=**Dummy** (p. ??)

Definition at line **56** of file **Dummy.java**.

6.11.3 Member Data Documentation

6.11.3.1 id

```
Long id [private]
```

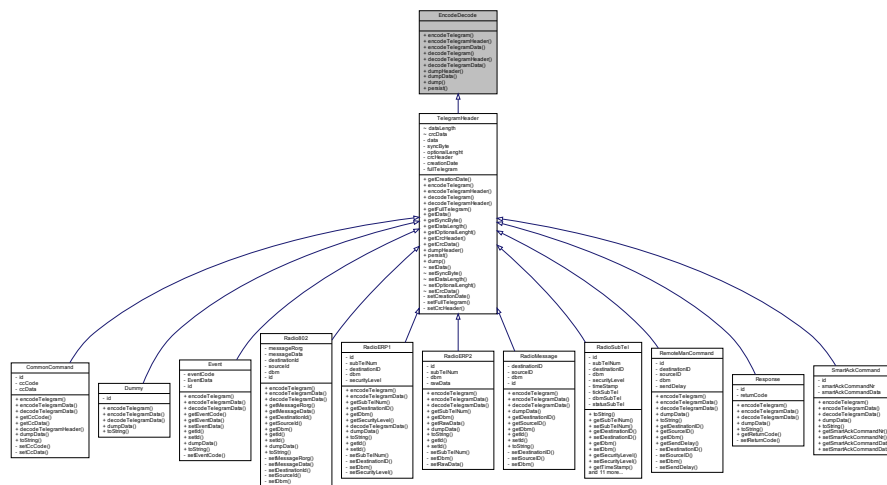
Definition at line **17** of file **Dummy.java**.

The documentation for this class was generated from the following file:

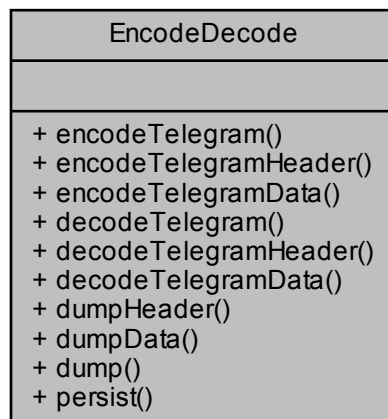
- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**Dummy.java**

6.12 EncodeDecode

Inheritance diagram for EncodeDecode:



Collaboration diagram for EncodeDecode:



Public Member Functions

- String **encodeTelegram** ()
encodeTelegram provides an example telegram hex for this packet type.
- String **encodeTelegramHeader** ()
encodeTelegramHeader provides an example header hex for this packet type.
- String **encodeTelegramData** ()
encodeTelegramData provides an example data hex for this packet type.

- void **decodeTelegram** (String hexTelegram)
decodeTelegram splits telegram in parts
- void **decodeTelegramHeader** (String hexTelegram)
decodeHeaderTelegram splits telegram header in parts
- void **decodeTelegramData** (String hexTelegram)
decodeTelegramData splits telegram data in parts
- void **dumpHeader** ()
dumpHeader Print Header, useful for logs and testing
- void **dumpData** ()
dumpData Print Data, useful for logs and testing EnOcean Telegram
- void **dump** ()
run both, the dumpHeader and the dumpData
- void **persist** ()
persist saves object to database

6.12.1 Detailed Description

Definition at line 7 of file **EncodeDecode.java**.

6.12.2 Member Function Documentation

6.12.2.1 decodeTelegram()

```
void decodeTelegram (
    String hexTelegram )
```

decodeTelegram splits telegram in parts

Parameters

<i>hexTelegram</i>	EnOcean Telegram
--------------------	------------------

Implemented in **TelegramHeader** (p. ??).

6.12.2.2 decodeTelegramData()

```
void decodeTelegramData (
    String hexTelegram )
```

decodeTelegramData splits telegram data in parts

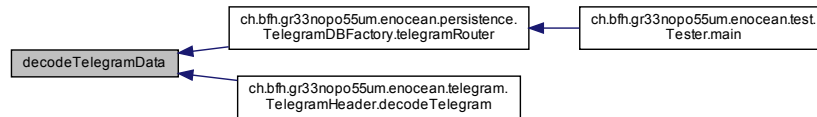
Parameters

<i>hexTelegram</i>	EnOcean Telegram
--------------------	------------------

Implemented in **RadioSubTel** (p. ??), **RadioERP1** (p. ??), **RadioERP2** (p. ??), **Radio802** (p. ??), **Response**

(p. ??), **RadioMessage** (p. ??), **RemoteManCommand** (p. ??), **Event** (p. ??), **SmartAckCommand** (p. ??), **CommonCommand** (p. ??), and **Dummy** (p. ??).

Here is the caller graph for this function:



6.12.2.3 decodeTelegramHeader()

```
void decodeTelegramHeader (
    String hexTelegram )
```

`decodeHeaderTelegram` splits telegram header in parts

Parameters

<i>hexTelegram</i>	EnOcean Telegram
--------------------	------------------

Implemented in **CommonCommand** (p. ??), and **TelegramHeader** (p. ??).

6.12.2.4 dump()

```
void dump ( )
```

run both, the `dumpHeader` and the `dumpData`

Implemented in **TelegramHeader** (p. ??).

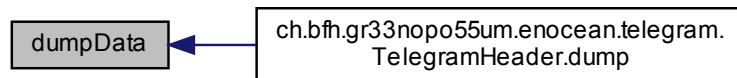
6.12.2.5 dumpData()

```
void dumpData ( )
```

`dumpData` Print Data, useful for logs and testing EnOcean Telegram

Implemented in **RadioSubTel** (p. ??), **Radio802** (p. ??), **RadioERP1** (p. ??), **RadioERP2** (p. ??), **Event** (p. ??), **CommonCommand** (p. ??), **RemoteManCommand** (p. ??), **RadioMessage** (p. ??), **Response** (p. ??), **SmartAckCommand** (p. ??), and **Dummy** (p. ??).

Here is the caller graph for this function:



6.12.2.6 dumpHeader()

```
void dumpHeader ( )
```

`dumpHeader` Print Header, useful for logs and testing

Implemented in **TelegramHeader** (p. ??).

6.12.2.7 encodeTelegram()

```
String encodeTelegram ( )
```

`encodeTelegram` provides an example telegram hex for this packet type.

Returns

String hexTelegram EnOcean Telegram

Implemented in **RadioSubTel** (p. ??), **TelegramHeader** (p. ??), **RadioERP2** (p. ??), **Radio802** (p. ??), **RadioERP1** (p. ??), **RemoteManCommand** (p. ??), **Response** (p. ??), **RadioMessage** (p. ??), **Event** (p. ??), **SmartAckCommand** (p. ??), **CommonCommand** (p. ??), and **Dummy** (p. ??).

6.12.2.8 encodeTelegramData()

```
String encodeTelegramData ( )
```

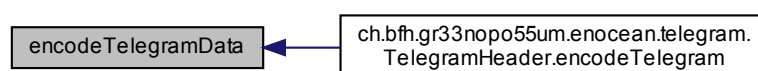
`encodeTelegramData` provides an example data hex for this packet type.

Returns

String hexTelegramData EnOcean Telegram

Implemented in **RadioSubTel** (p. ??), **RadioERP2** (p. ??), **Radio802** (p. ??), **RadioERP1** (p. ??), **Response** (p. ??), **RadioMessage** (p. ??), **RemoteManCommand** (p. ??), **Event** (p. ??), **SmartAckCommand** (p. ??), **CommonCommand** (p. ??), and **Dummy** (p. ??).

Here is the caller graph for this function:



6.12.2.9 encodeTelegramHeader()

```
String encodeTelegramHeader ( )
```

encodeTelegramHeader provides an example header hex for this packet type.

Returns

String hexTelegramHeader EnOcean Telegram

Implemented in **TelegramHeader** (p. ??).

6.12.2.10 persist()

```
void persist ( )
```

persist saves object to database

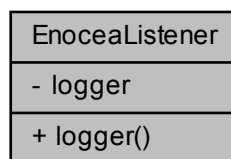
Implemented in **TelegramHeader** (p. ??).

The documentation for this interface was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**EncodeDecode.java**

6.13 EnocleaListener

Collaboration diagram for EnocleaListener:



Public Member Functions

- void **logger** ()
Logger for Serial port.

Static Private Attributes

- static Logger **logger** = LogManager.getLogger()

6.13.1 Detailed Description

Definition at line 16 of file **EnocearLayouter.java**.

6.13.2 Member Function Documentation

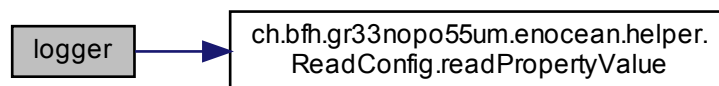
6.13.2.1 logger()

```
void logger ( )
```

Logger for Serial port.

Definition at line 24 of file **EnocearLayouter.java**.

Here is the call graph for this function:



6.13.3 Member Data Documentation

6.13.3.1 logger

```
Logger logger = LogManager.getLogger() [static], [private]
```

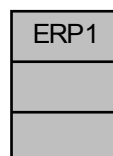
Definition at line 18 of file **EnocearLayouter.java**.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoclean/application/**EnocearLayouter.java**

6.14 ERP1

Collaboration diagram for ERP1:



6.14.1 Detailed Description

Author

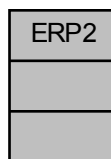
silas , for further informations check EnOceanSerialProtocol v3. **This** (p. ??) is usually the default telegram type.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**RadioERP1.java**

6.15 ERP2

Collaboration diagram for ERP2:



6.15.1 Detailed Description

telegram, for further informations check EnOceanSerialProtocol v3

Author

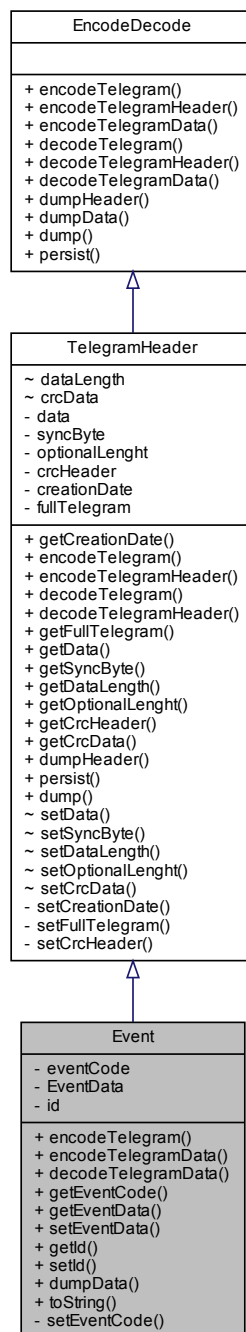
silas & louis

The documentation for this class was generated from the following file:

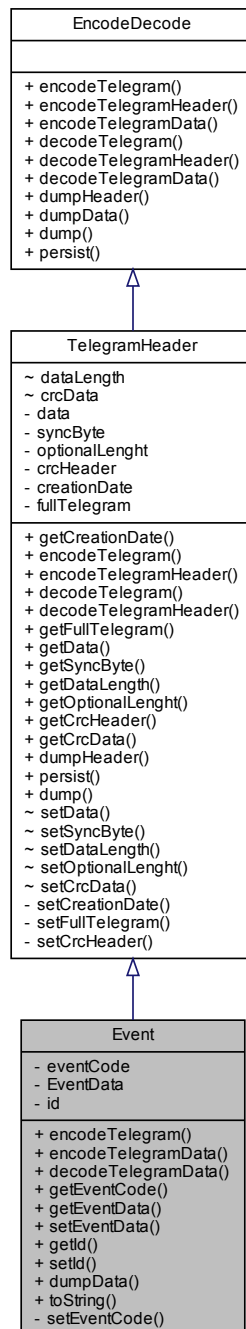
- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**RadioERP2.java**

6.16 Event

Inheritance diagram for Event:



Collaboration diagram for Event:



Public Member Functions

- String **encodeTelegram** ()
encodeTelegram provides an example telegram hex for this packet type.
- String **encodeTelegramData** ()
encodeTelegramData provides an example data hex for this packet type.
- void **decodeTelegramData** (String hexTelegram)

decodeTelegramData splits telegram in parts

- int **getEventCode** ()
 - int **getEventData** ()
 - void **setEventData** (int eventData)
 - Long **getId** ()
 - void **setId** (Long id)
 - void **dumpData** ()
- dumpData Print Data, useful for logs and testing*
- String **toString** ()

Private Member Functions

- void **setEventCode** (int eventCode)

Private Attributes

- int **eventCode**
- int **EventData**
- Long **id**

Additional Inherited Members

6.16.1 Detailed Description

Definition at line 13 of file **Event.java**.

6.16.2 Member Function Documentation

6.16.2.1 decodeTelegramData()

```
void decodeTelegramData (
    String hexTelegram )
```

decodeTelegramData splits telegram in parts

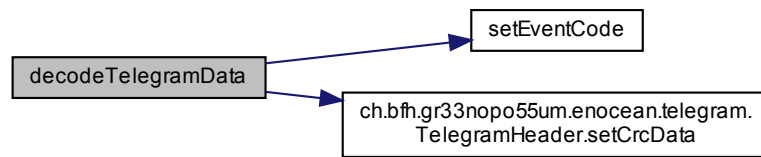
Parameters

<i>hexTelegram</i>	in hex String
--------------------	---------------

Implements **EncodeDecode** (p. ??).

Definition at line 46 of file **Event.java**.

Here is the call graph for this function:



6.16.2.2 dumpData()

```
void dumpData ( )
```

`dumpData` Print Data, useful for logs and testing

Implements **EncodeDecode** (p. ??).

Definition at line **83** of file **Event.java**.

Here is the call graph for this function:



6.16.2.3 encodeTelegram()

```
String encodeTelegram ( )
```

`encodeTelegram` provides an example telegram hex for this packet type.

Returns

telegram hex

Implements **EncodeDecode** (p. ??).

Definition at line **28** of file **Event.java**.

6.16.2.4 encodeTelegramData()

```
String encodeTelegramData ( )
```

encodeTelegramData provides an example data hex for this packet type.

Returns

String hexTelegramData EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line **37** of file **Event.java**.

6.16.2.5 getEventCode()

```
int getEventCode ( )
```

Definition at line **55** of file **Event.java**.

6.16.2.6 getEventData()

```
int getEventData ( )
```

Definition at line **63** of file **Event.java**.

6.16.2.7 getId()

```
Long getId ( )
```

Definition at line **71** of file **Event.java**.

6.16.2.8 setEventCode()

```
void setEventCode (
    int eventCode ) [private]
```

Definition at line **59** of file **Event.java**.

Here is the caller graph for this function:



6.16.2.9 setEventData()

```
void setEventData (
    int eventData )
```

Definition at line **67** of file **Event.java**.

6.16.2.10 setId()

```
void setId (
    Long id )
```

Definition at line **75** of file **Event.java**.

6.16.2.11 toString()

```
String toString ( )
```

Definition at line **89** of file **Event.java**.

Here is the caller graph for this function:



6.16.3 Member Data Documentation

6.16.3.1 eventCode

```
int eventCode [private]
```

Definition at line **15** of file **Event.java**.

6.16.3.2 EventData

```
int EventData [private]
```

Definition at line **16** of file **Event.java**.

6.16.3.3 id

```
Long id [private]
```

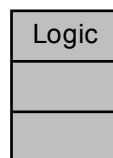
Definition at line 20 of file **Event.java**.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**Event.java**

6.17 Logic

Collaboration diagram for Logic:



6.17.1 Detailed Description

Author

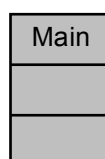
silas which packettype should be persisted.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/persistence/**TelegramDBFactory.java**↔

6.18 Main

Collaboration diagram for Main:



6.18.1 Detailed Description

Author

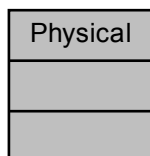
silas

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/application/**Application.java**

6.19 Physical

Collaboration diagram for Physical:



6.19.1 Detailed Description

Author

silas & louis , for further informations check EnOceanSerialProtocol v3

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**Radio802.java**

6.20 Radio

Collaboration diagram for Radio:



6.20.1 Detailed Description

data without any radio telegram contents, for further informations check EnOceanSerialProtocol v3

Author

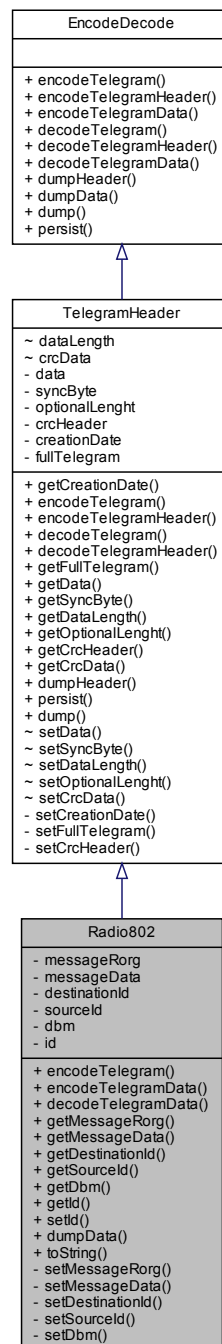
silas & louis

The documentation for this class was generated from the following file:

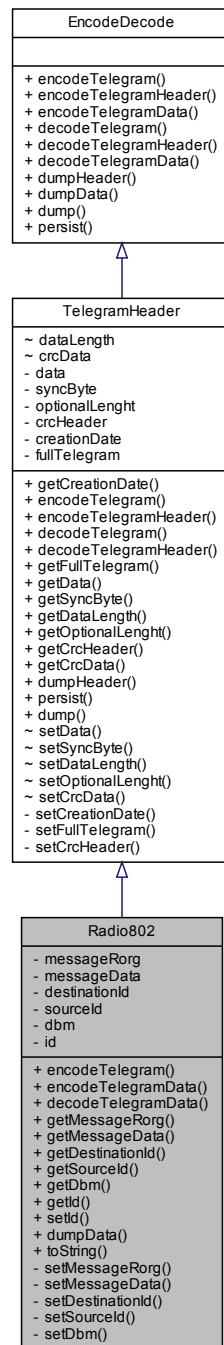
- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/RadioMessage.java`

6.21 Radio802

Inheritance diagram for Radio802:



Collaboration diagram for Radio802:



Public Member Functions

- String **encodeTelegram** ()
encodeTelegram provides an example telegram hex for this packet type.
- String **encodeTelegramData** ()
encodeTelegramData provides an example data hex for this packet type.
- void **decodeTelegramData** (String hexTelegram)

decodeTelegramData splits telegram in parts

- int **getMessageRorg** ()
 - int **getMessageData** ()
 - int **getDestinationId** ()
 - int **getSourceId** ()
 - int **getDbm** ()
 - Long **getId** ()
 - void **setId** (Long **id**)
 - void **dumpData** ()
- dumpData Print Data, useful for logs and testiing*
- String **toString** ()

Private Member Functions

- void **setMessageRorg** (int **messageRorg**)
- void **setMessageData** (int **messageData**)
- void **setDestinationId** (int **destinationId**)
- void **setSourceId** (int **sourceId**)
- void **setDbm** (int **dbm**)

Private Attributes

- int **messageRorg**
- int **messageData**
- int **destinationId**
- int **sourceId**
- int **dbm**
- Long **id**

Additional Inherited Members

6.21.1 Detailed Description

Definition at line 13 of file **Radio802.java**.

6.21.2 Member Function Documentation

6.21.2.1 decodeTelegramData()

```
void decodeTelegramData (
    String hexTelegram )
```

decodeTelegramData splits telegram in parts

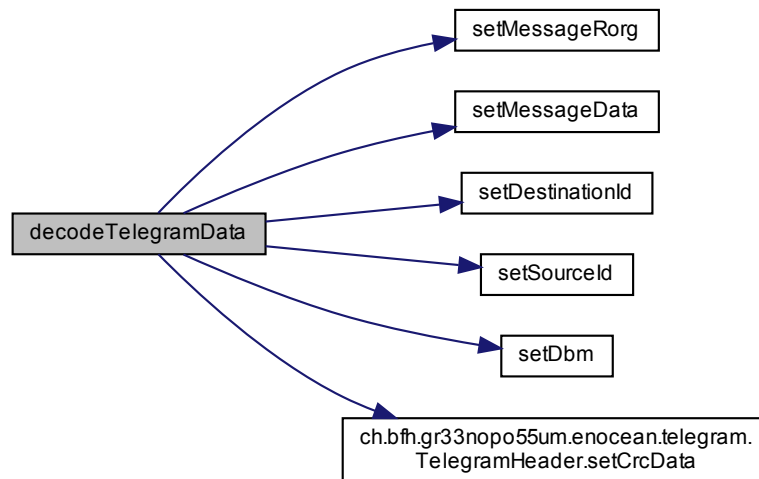
Parameters

<i>hexTelegram</i>	in hex String
--------------------	---------------

Implements **EncodeDecode** (p. ??).

Definition at line **50** of file **Radio802.java**.

Here is the call graph for this function:



6.21.2.2 dumpData()

```
void dumpData ( )
```

`dumpData` Print Data, useful for logs and testiing

Implements **EncodeDecode** (p. ??).

Definition at line **114** of file **Radio802.java**.

Here is the call graph for this function:



6.21.2.3 encodeTelegram()

```
String encodeTelegram ( )
```

encodeTelegram provides an example telegram hex for this packet type.

Returns

telegram hex

Implements **EncodeDecode** (p. ??).

Definition at line **32** of file **Radio802.java**.

6.21.2.4 encodeTelegramData()

```
String encodeTelegramData ( )
```

encodeTelegramData provides an example data hex for this packet type.

Returns

String hexTelegramData EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line **41** of file **Radio802.java**.

6.21.2.5 getDbm()

```
int getDbm ( )
```

Definition at line **94** of file **Radio802.java**.

6.21.2.6 getDestinationId()

```
int getDestinationId ( )
```

Definition at line **78** of file **Radio802.java**.

6.21.2.7 getId()

```
Long getId ( )
```

Definition at line **102** of file **Radio802.java**.

6.21.2.8 getMessageData()

```
int getMessageData ( )
```

Definition at line 70 of file **Radio802.java**.

6.21.2.9 getMessageRorg()

```
int getMessageRorg ( )
```

Definition at line 62 of file **Radio802.java**.

6.21.2.10 getSourceId()

```
int getSourceId ( )
```

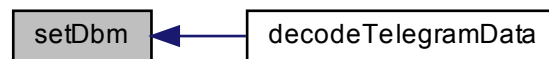
Definition at line 86 of file **Radio802.java**.

6.21.2.11 setDbm()

```
void setDbm (
    int dbm ) [private]
```

Definition at line 98 of file **Radio802.java**.

Here is the caller graph for this function:



6.21.2.12 setDestinationId()

```
void setDestinationId (
    int destinationId ) [private]
```

Definition at line 82 of file **Radio802.java**.

Here is the caller graph for this function:



6.21.2.13 setId()

```
void setId (
    Long id )
```

Definition at line **106** of file **Radio802.java**.

6.21.2.14 setMessageData()

```
void setMessageData (
    int messageData ) [private]
```

Definition at line **74** of file **Radio802.java**.

Here is the caller graph for this function:



6.21.2.15 setMessageRorg()

```
void setMessageRorg (
    int messageRorg ) [private]
```

Definition at line **66** of file **Radio802.java**.

Here is the caller graph for this function:

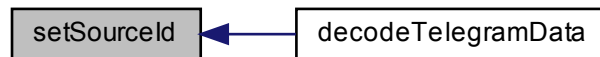


6.21.2.16 setSourceId()

```
void setSourceId (  
    int sourceId ) [private]
```

Definition at line **90** of file **Radio802.java**.

Here is the caller graph for this function:



6.21.2.17 toString()

```
String toString ( )
```

Definition at line **124** of file **Radio802.java**.

Here is the caller graph for this function:



6.21.3 Member Data Documentation

6.21.3.1 dbm

```
int dbm [private]
```

Definition at line **21** of file **Radio802.java**.

6.21.3.2 destinationId

```
int destinationId [private]
```

Definition at line **18** of file **Radio802.java**.

6.21.3.3 id

```
Long id [private]
```

Definition at line **24** of file **Radio802.java**.

6.21.3.4 messageData

```
int messageData [private]
```

Definition at line **16** of file **Radio802.java**.

6.21.3.5 messageRorg

```
int messageRorg [private]
```

Definition at line **15** of file **Radio802.java**.

6.21.3.6 sourceId

```
int sourceId [private]
```

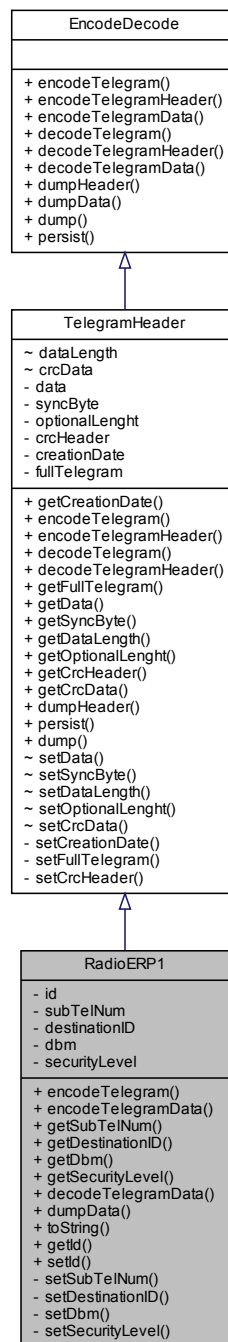
Definition at line **20** of file **Radio802.java**.

The documentation for this class was generated from the following file:

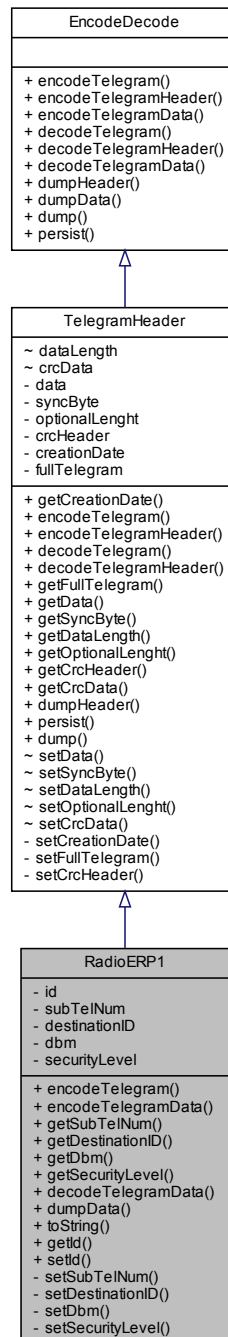
- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**Radio802.java**

6.22 RadioERP1

Inheritance diagram for RadioERP1:



Collaboration diagram for RadioERP1:



Public Member Functions

- String **encodeTelegram** ()
encodeTelegram provides an example telegram hex for this packet type.
- String **encodeTelegramData** ()
encodeTelegramData provides an example data hex for this packet type.
- int **getSubTelNum** ()

- String **getDestinationID** ()
- int **getDbm** ()
- int **getSecurityLevel** ()
- void **decodeTelegramData** (String hexTelegram)
decodeTelegramData splits telegram in parts
- void **dumpData** ()
dumpData Print Data, useful for logs and testing EnOcean Telegram
- String **toString** ()
- Long **getId** ()
- void **setId** (Long id)

Private Member Functions

- void **setSubTelNum** (int subTelNum)
- void **setDestinationID** (String destinationID)
- void **setDbm** (int dbm)
- void **setSecurityLevel** (int securityLevel)

Private Attributes

- Long **id**
- int **subTelNum**
- String **destinationID**
- int **dbm**
- int **securityLevel**

Additional Inherited Members

6.22.1 Detailed Description

Definition at line 14 of file **RadioERP1.java**.

6.22.2 Member Function Documentation

6.22.2.1 decodeTelegramData()

```
void decodeTelegramData (
    String hexTelegram )
```

decodeTelegramData splits telegram in parts

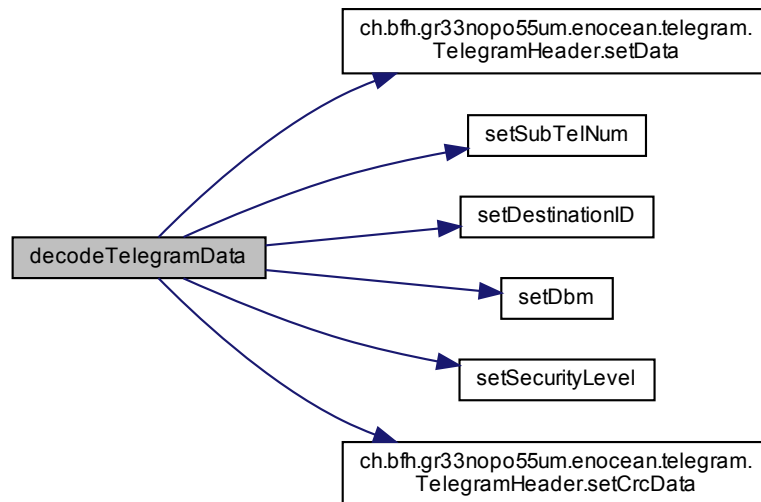
Parameters

<i>hexTelegram</i>	in hex String
--------------------	---------------

Implements **EncodeDecode** (p. ??).

Definition at line 81 of file **RadioERP1.java**.

Here is the call graph for this function:



6.22.2.2 dumpData()

```
void dumpData ( )
```

`dumpData` Print Data, useful for logs and testing EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line 98 of file **RadioERP1.java**.

Here is the call graph for this function:



6.22.2.3 encodeTelegram()

```
String encodeTelegram ( )
```

encodeTelegram provides an example telegram hex for this packet type.

Returns

telegram hex

Implements **EncodeDecode** (p. ??).

Definition at line **31** of file **RadioERP1.java**.

6.22.2.4 encodeTelegramData()

```
String encodeTelegramData ( )
```

encodeTelegramData provides an example data hex for this packet type.

Returns

String hexTelegramData EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line **40** of file **RadioERP1.java**.

6.22.2.5 getDbm()

```
int getDbm ( )
```

Definition at line **60** of file **RadioERP1.java**.

6.22.2.6 getDestinationID()

```
String getDestinationID ( )
```

Definition at line **52** of file **RadioERP1.java**.

6.22.2.7 getId()

```
Long getId ( )
```

Definition at line **114** of file **RadioERP1.java**.

6.22.2.8 `getSecurityLevel()`

```
int getSecurityLevel ( )
```

Definition at line **68** of file **RadioERP1.java**.

6.22.2.9 `getSubTelNum()`

```
int getSubTelNum ( )
```

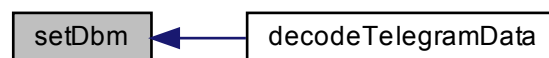
Definition at line **44** of file **RadioERP1.java**.

6.22.2.10 `setDbm()`

```
void setDbm (
    int dbm ) [private]
```

Definition at line **64** of file **RadioERP1.java**.

Here is the caller graph for this function:



6.22.2.11 `setDestinationID()`

```
void setDestinationID (
    String destinationID ) [private]
```

Definition at line **56** of file **RadioERP1.java**.

Here is the caller graph for this function:



6.22.2.12 setId()

```
void setId (
    Long id )
```

Definition at line **118** of file **RadioERP1.java**.

6.22.2.13 setSecurityLevel()

```
void setSecurityLevel (
    int securityLevel ) [private]
```

Definition at line **72** of file **RadioERP1.java**.

Here is the caller graph for this function:



6.22.2.14 setSubTelNum()

```
void setSubTelNum (
    int subTelNum ) [private]
```

Definition at line **48** of file **RadioERP1.java**.

Here is the caller graph for this function:



6.22.2.15 toString()

```
String toString ( )
```

Definition at line **105** of file **RadioERP1.java**.

Here is the caller graph for this function:



6.22.3 Member Data Documentation

6.22.3.1 dbm

```
int dbm [private]
```

Definition at line **22** of file **RadioERP1.java**.

6.22.3.2 destinationID

```
String destinationID [private]
```

Definition at line **21** of file **RadioERP1.java**.

6.22.3.3 id

```
Long id [private]
```

Definition at line **18** of file **RadioERP1.java**.

6.22.3.4 securityLevel

```
int securityLevel [private]
```

Definition at line **23** of file **RadioERP1.java**.

6.22.3.5 subTelNum

```
int subTelNum [private]
```

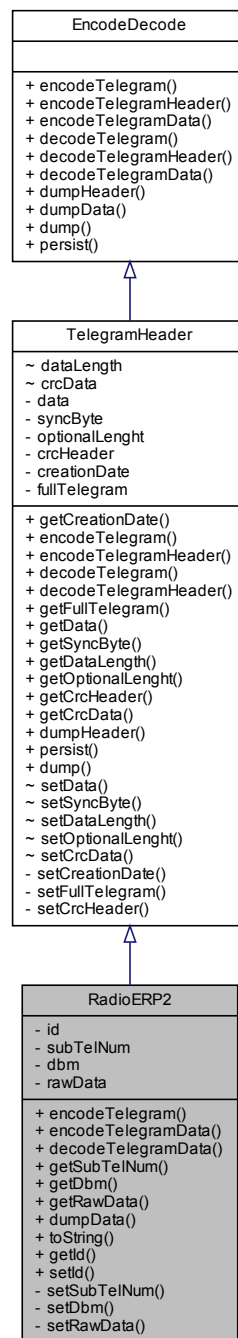
Definition at line 20 of file **RadioERP1.java**.

The documentation for this class was generated from the following file:

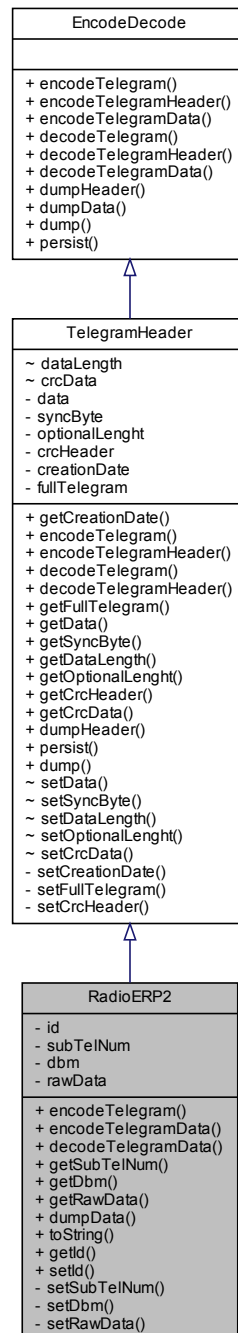
- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**RadioERP1.java**

6.23 RadioERP2

Inheritance diagram for RadioERP2:



Collaboration diagram for RadioERP2:



Public Member Functions

- String **encodeTelegram** ()
encodeTelegram provides an example telegram hex for this packet type.
- String **encodeTelegramData** ()
encodeTelegramData provides an example data hex for this packet type.
- void **decodeTelegramData** (String hexTelegram)

decodeTelegramData splits telegram in parts

- int **getSubTelNum** ()
- int **getDbm** ()
- String **getRawData** ()
- void **dumpData** ()

dumpData Print Data, useful for logs and testing EnOcean Telegram

- String **toString** ()
- Long **getId** ()
- void **setId** (Long id)

Private Member Functions

- void **setSubTelNum** (int subTelNum)
- void **setDbm** (int dbm)
- void **setRawData** (String rawData)

Private Attributes

- Long **id**
- int **subTelNum**
- int **dbm**
- String **rawData**

Additional Inherited Members

6.23.1 Detailed Description

Definition at line 14 of file **RadioERP2.java**.

6.23.2 Member Function Documentation

6.23.2.1 decodeTelegramData()

```
void decodeTelegramData (
    String hexTelegram )
```

decodeTelegramData splits telegram in parts

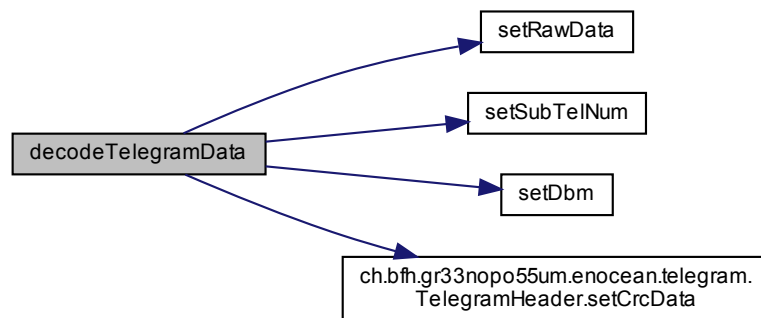
Parameters

<i>hexTelegram</i>	in hex String
--------------------	---------------

Implements **EncodeDecode** (p. ??).

Definition at line 53 of file **RadioERP2.java**.

Here is the call graph for this function:



6.23.2.2 dumpData()

```
void dumpData ( )
```

`dumpData` Print Data, useful for logs and testing EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line **92** of file **RadioERP2.java**.

Here is the call graph for this function:



6.23.2.3 encodeTelegram()

```
String encodeTelegram ( )
```

`encodeTelegram` provides an example telegram hex for this packet type.

Returns

telegram hex

Implements **EncodeDecode** (p. ??).

Definition at line **33** of file **RadioERP2.java**.

6.23.2.4 encodeTelegramData()

```
String encodeTelegramData ( )
```

encodeTelegramData provides an example data hex for this packet type.

Returns

String hexTelegramData EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line **43** of file **RadioERP2.java**.

6.23.2.5 getDbm()

```
int getDbm ( )
```

Definition at line **74** of file **RadioERP2.java**.

6.23.2.6 getId()

```
Long getId ( )
```

Definition at line **106** of file **RadioERP2.java**.

6.23.2.7 getRawData()

```
String getRawData ( )
```

Definition at line **82** of file **RadioERP2.java**.

6.23.2.8 getSubTelNum()

```
int getSubTelNum ( )
```

Returns

subTelNum

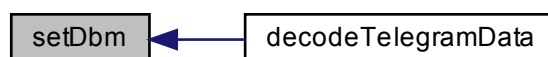
Definition at line **66** of file **RadioERP2.java**.

6.23.2.9 setDbm()

```
void setDbm (
    int dbm ) [private]
```

Definition at line **78** of file **RadioERP2.java**.

Here is the caller graph for this function:



6.23.2.10 setId()

```
void setId (
    Long id )
```

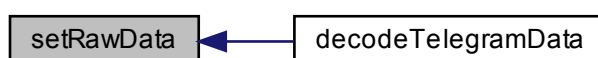
Definition at line **110** of file **RadioERP2.java**.

6.23.2.11 setRawData()

```
void setRawData (
    String rawData ) [private]
```

Definition at line **86** of file **RadioERP2.java**.

Here is the caller graph for this function:



6.23.2.12 setSubTelNum()

```
void setSubTelNum (
    int subTelNum ) [private]
```

Definition at line **70** of file **RadioERP2.java**.

Here is the caller graph for this function:



6.23.2.13 toString()

```
String toString ( )
```

Definition at line **99** of file **RadioERP2.java**.

Here is the caller graph for this function:



6.23.3 Member Data Documentation

6.23.3.1 dbm

```
int dbm [private]
```

Definition at line **23** of file **RadioERP2.java**.

6.23.3.2 id

```
Long id [private]
```

Definition at line **19** of file **RadioERP2.java**.

6.23.3.3 rawData

```
String rawData [private]
```

Definition at line 24 of file **RadioERP2.java**.

6.23.3.4 subTelNum

```
int subTelNum [private]
```

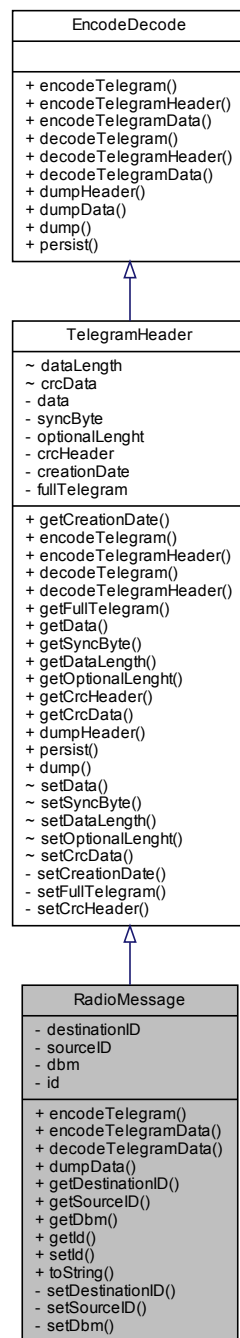
Definition at line 22 of file **RadioERP2.java**.

The documentation for this class was generated from the following file:

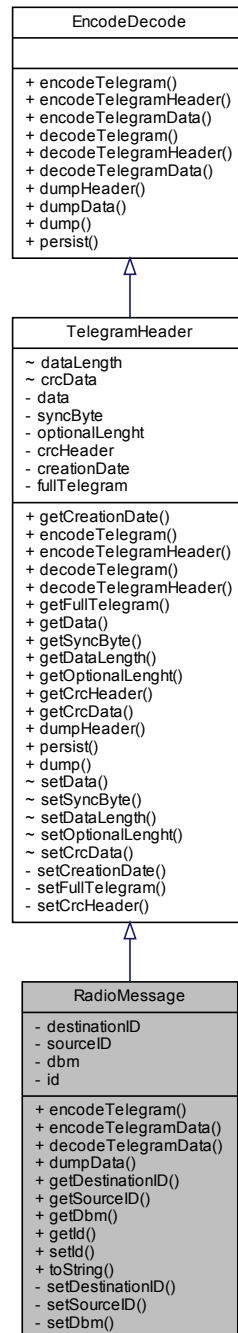
- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**RadioERP2.java**

6.24 RadioMessage

Inheritance diagram for RadioMessage:



Collaboration diagram for RadioMessage:



Public Member Functions

- String **encodeTelegram** ()
- String **encodeTelegramData** ()
- void **decodeTelegramData** (String hexTelegram)
- void **dumpData** ()
- String **getDestinationID** ()

- String **getSourceID** ()
- int **getDbm** ()
- Long **getId** ()
- void **setId** (Long **id**)
- String **toString** ()

Private Member Functions

- void **setDestinationID** (String **destinationID**)
- void **setSourceID** (String **sourceID**)
- void **setDbm** (int **dbm**)

Private Attributes

- String **destinationID**
- String **sourceID**
- int **dbm**
- Long **id**

Additional Inherited Members

6.24.1 Detailed Description

Definition at line **14** of file **RadioMessage.java**.

6.24.2 Member Function Documentation

6.24.2.1 decodeTelegramData()

```
void decodeTelegramData (
    String hexTelegram )
```

decodeTelegramData splits telegram in parts

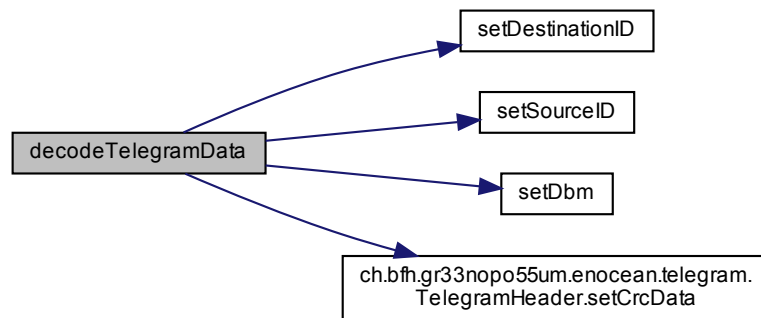
Parameters

<i>hexTelegram</i>	in hex String
--------------------	---------------

Implements **EncodeDecode** (p. ??).

Definition at line **49** of file **RadioMessage.java**.

Here is the call graph for this function:



6.24.2.2 dumpData()

```
void dumpData ( )
```

`dumpData` Print Data, useful for logs and testiing

EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line **63** of file **RadioMessage.java**.

Here is the call graph for this function:



6.24.2.3 encodeTelegram()

```
String encodeTelegram ( )
```

`encodeTelegram` provides an example telegram hex for this packet type.

Returns

telegram hex

Implements **EncodeDecode** (p. ??).

Definition at line **29** of file **RadioMessage.java**.

6.24.2.4 encodeTelegramData()

```
String encodeTelegramData ( )
```

encodeTelegramData provides an example data hex for this packet type.

Returns

String hexTelegramData EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line **39** of file **RadioMessage.java**.

6.24.2.5 getDbm()

```
int getDbm ( )
```

Definition at line **84** of file **RadioMessage.java**.

6.24.2.6 getDestinationID()

```
String getDestinationID ( )
```

Definition at line **68** of file **RadioMessage.java**.

6.24.2.7 getId()

```
Long getId ( )
```

Definition at line **92** of file **RadioMessage.java**.

6.24.2.8 getSourceID()

```
String getSourceID ( )
```

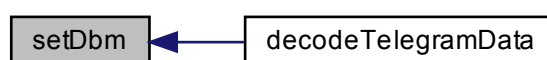
Definition at line **76** of file **RadioMessage.java**.

6.24.2.9 setDbm()

```
void setDbm (
    int dbm ) [private]
```

Definition at line **88** of file **RadioMessage.java**.

Here is the caller graph for this function:



6.24.2.10 setDestinationID()

```
void setDestinationID (
    String destinationID ) [private]
```

Definition at line **72** of file **RadioMessage.java**.

Here is the caller graph for this function:



6.24.2.11 setId()

```
void setId (
    Long id )
```

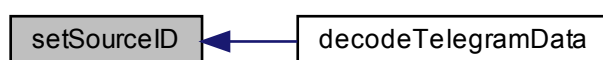
Definition at line **96** of file **RadioMessage.java**.

6.24.2.12 setSourceID()

```
void setSourceID (
    String sourceID ) [private]
```

Definition at line **80** of file **RadioMessage.java**.

Here is the caller graph for this function:



6.24.2.13 toString()

```
String toString ( )
```

Definition at line **101** of file **RadioMessage.java**.

Here is the caller graph for this function:



6.24.3 Member Data Documentation

6.24.3.1 dbm

```
int dbm [private]
```

Definition at line **17** of file **RadioMessage.java**.

6.24.3.2 destinationID

```
String destinationID [private]
```

Definition at line **15** of file **RadioMessage.java**.

6.24.3.3 id

```
Long id [private]
```

Definition at line **21** of file **RadioMessage.java**.

6.24.3.4 sourceID

```
String sourceID [private]
```

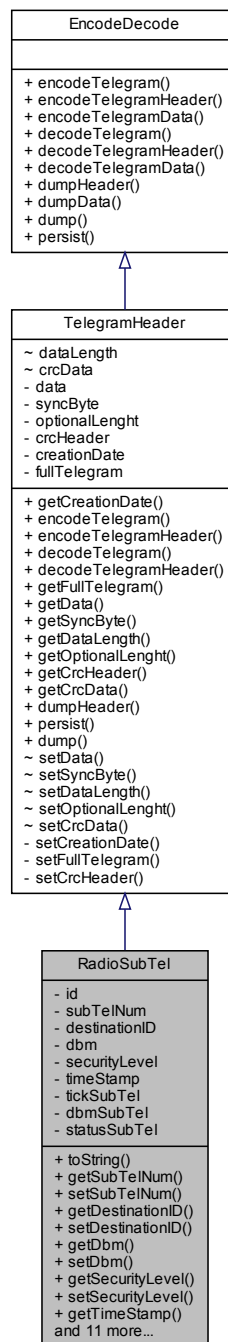
Definition at line **16** of file **RadioMessage.java**.

The documentation for this class was generated from the following file:

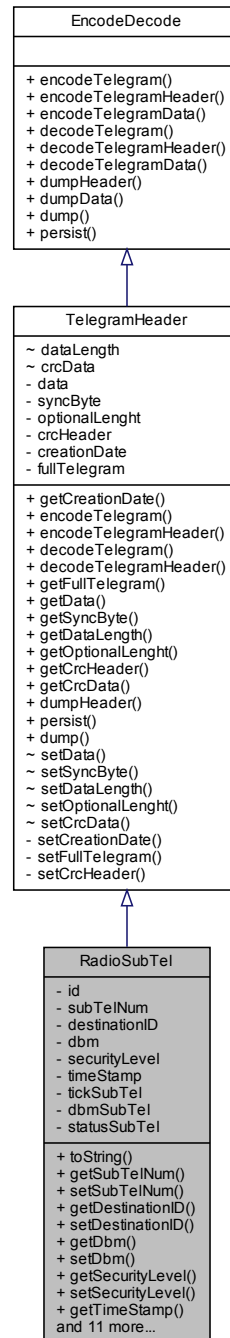
- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/RadioMessage.java`

6.25 RadioSubTel

Inheritance diagram for RadioSubTel:



Collaboration diagram for RadioSubTel:



Public Member Functions

- String **toString** ()
- int **getSubTelNum** ()
- void **setSubTelNum** (int **subTelNum**)
- String **getDestinationID** ()
- void **setDestinationID** (String **destinationID**)

- int **getDbm** ()
- void **setDbm** (int dbm)
- int **getSecurityLevel** ()
- void **setSecurityLevel** (int securityLevel)
- String **getTimeStamp** ()
- void **setTimeStamp** (String timeStamp)
- String **getTickSubTel** ()
- void **setTickSubTel** (String tickSubTel)
- String **getDbmSubTel** ()
- void **setDbmSubTel** (String dbmSubTel)
- String **getStatusSubTel** ()
- void **setStatusSubTel** (String statusSubTel)
- String **encodeTelegram** ()
 - encodeTelegram provides an example telegram hex for this packet type.*
- String **encodeTelegramData** ()
 - encodeTelegramData provides an example data hex for this packet type.*
- void **decodeTelegramData** (String hexTelegram)
 - decodeTelegramData splits telegram in parts*
- void **dumpData** ()
 - dumpData Print Data, useful for logs and testiing*

Private Attributes

- Long **id**
- int **subTelNum**
- String **destinationID**
- int **dbm**
- int **securityLevel**
- String **timeStamp**
- String **tickSubTel**
- String **dbmSubTel**
- String **statusSubTel**

Additional Inherited Members

6.25.1 Detailed Description

Definition at line 14 of file **RadioSubTel.java**.

6.25.2 Member Function Documentation

6.25.2.1 decodeTelegramData()

```
void decodeTelegramData (
    String hexTelegram )
```

decodeTelegramData splits telegram in parts

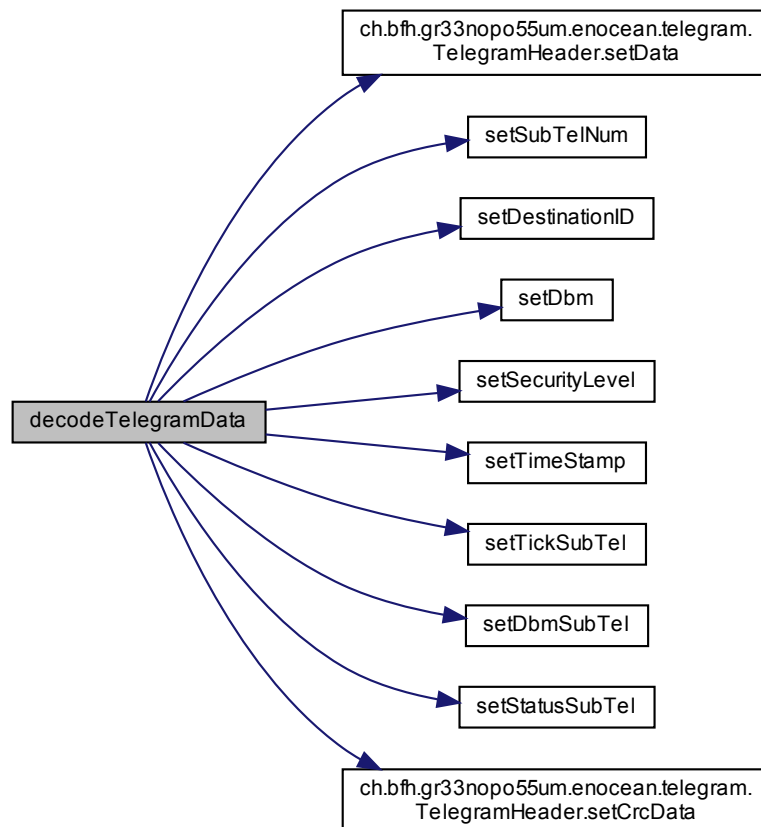
Parameters

<i>hexTelegram</i>	
--------------------	--

Implements **EncodeDecode** (p. ??).

Definition at line **131** of file **RadioSubTel.java**.

Here is the call graph for this function:



6.25.2.2 dumpData()

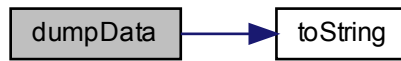
```
void dumpData ( )
```

`dumpData` Print Data, useful for logs and testiing

Implements **EncodeDecode** (p. ??).

Definition at line **160** of file **RadioSubTel.java**.

Here is the call graph for this function:



6.25.2.3 encodeTelegram()

```
String encodeTelegram ( )
```

`encodeTelegram` provides an example telegram hex for this packet type.

Returns

telegram hex

Implements **EncodeDecode** (p. ??).

Definition at line **112** of file **RadioSubTel.java**.

6.25.2.4 encodeTelegramData()

```
String encodeTelegramData ( )
```

`encodeTelegramData` provides an example data hex for this packet type.

Returns

String hexTelegramData EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line **121** of file **RadioSubTel.java**.

6.25.2.5 getDbm()

```
int getDbm ( )
```

Definition at line **59** of file **RadioSubTel.java**.

6.25.2.6 getDbmSubTel()

```
String getDbmSubTel ( )
```

Definition at line **91** of file **RadioSubTel.java**.

6.25.2.7 getDestinationID()

```
String getDestinationID ( )
```

Definition at line 51 of file **RadioSubTel.java**.

6.25.2.8 getSecurityLevel()

```
int getSecurityLevel ( )
```

Definition at line 67 of file **RadioSubTel.java**.

6.25.2.9 getStatusSubTel()

```
String getStatusSubTel ( )
```

Definition at line 99 of file **RadioSubTel.java**.

6.25.2.10 getSubTelNum()

```
int getSubTelNum ( )
```

Definition at line 43 of file **RadioSubTel.java**.

6.25.2.11 getTickSubTel()

```
String getTickSubTel ( )
```

Definition at line 83 of file **RadioSubTel.java**.

6.25.2.12 getTimeStamp()

```
String getTimeStamp ( )
```

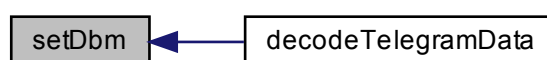
Definition at line 75 of file **RadioSubTel.java**.

6.25.2.13 setDbm()

```
void setDbm (  
    int dbm )
```

Definition at line 63 of file **RadioSubTel.java**.

Here is the caller graph for this function:

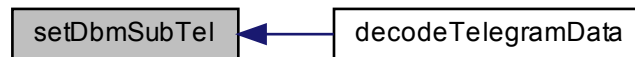


6.25.2.14 setDbmSubTel()

```
void setDbmSubTel (
    String dbmSubTel )
```

Definition at line **95** of file **RadioSubTel.java**.

Here is the caller graph for this function:



6.25.2.15 setDestinationID()

```
void setDestinationID (
    String destinationID )
```

Definition at line **55** of file **RadioSubTel.java**.

Here is the caller graph for this function:



6.25.2.16 setSecurityLevel()

```
void setSecurityLevel (
    int securityLevel )
```

Definition at line **71** of file **RadioSubTel.java**.

Here is the caller graph for this function:



6.25.2.17 setStatusSubTel()

```
void setStatusSubTel (
    String statusSubTel )
```

Definition at line **103** of file **RadioSubTel.java**.

Here is the caller graph for this function:



6.25.2.18 setSubTelNum()

```
void setSubTelNum (
    int subTelNum )
```

Definition at line **47** of file **RadioSubTel.java**.

Here is the caller graph for this function:



6.25.2.19 setTickSubTel()

```
void setTickSubTel (
    String tickSubTel )
```

Definition at line **87** of file **RadioSubTel.java**.

Here is the caller graph for this function:

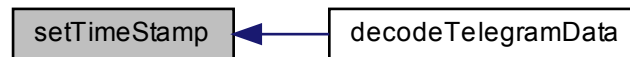


6.25.2.20 setTimeStamp()

```
void setTimeStamp (
    String timeStamp )
```

Definition at line **79** of file **RadioSubTel.java**.

Here is the caller graph for this function:



6.25.2.21 toString()

```
String toString ( )
```

Definition at line **31** of file **RadioSubTel.java**.

Here is the caller graph for this function:



6.25.3 Member Data Documentation

6.25.3.1 dbm

```
int dbm [private]
```

Definition at line **23** of file **RadioSubTel.java**.

6.25.3.2 dbmSubTel

```
String dbmSubTel [private]
```

Definition at line **27** of file **RadioSubTel.java**.

6.25.3.3 destinationID

```
String destinationID [private]
```

Definition at line 22 of file **RadioSubTel.java**.

6.25.3.4 id

```
Long id [private]
```

Definition at line 20 of file **RadioSubTel.java**.

6.25.3.5 securityLevel

```
int securityLevel [private]
```

Definition at line 24 of file **RadioSubTel.java**.

6.25.3.6 statusSubTel

```
String statusSubTel [private]
```

Definition at line 28 of file **RadioSubTel.java**.

6.25.3.7 subTelNum

```
int subTelNum [private]
```

Definition at line 21 of file **RadioSubTel.java**.

6.25.3.8 tickSubTel

```
String tickSubTel [private]
```

Definition at line 26 of file **RadioSubTel.java**.

6.25.3.9 timeStamp

```
String timeStamp [private]
```

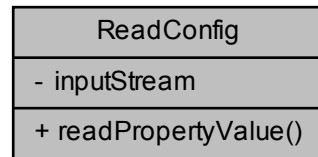
Definition at line 25 of file **RadioSubTel.java**.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**RadioSubTel.java**

6.26 ReadConfig

Collaboration diagram for ReadConfig:



Public Member Functions

- String **readPropertyValue** (String value) throws IOException

Private Attributes

- InputStream **inputStream**

6.26.1 Detailed Description

Definition at line 13 of file **ReadConfig.java**.

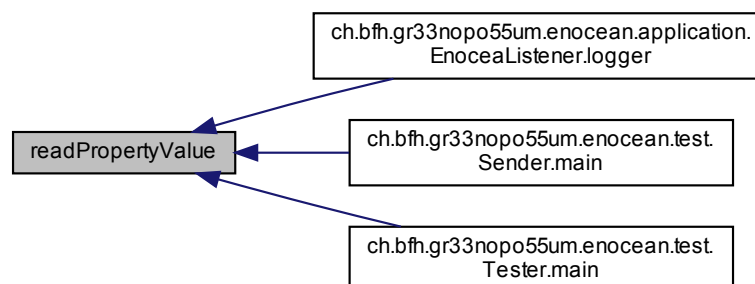
6.26.2 Member Function Documentation

6.26.2.1 readPropertyValue()

```
String readPropertyValue (
    String value ) throws IOException
```

Definition at line 16 of file **ReadConfig.java**.

Here is the caller graph for this function:



6.26.3 Member Data Documentation

6.26.3.1 inputStream

```
InputStream inputStream [private]
```

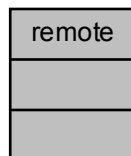
Definition at line **14** of file **ReadConfig.java**.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/**ReadConfig.java**

6.27 remote

Collaboration diagram for remote:



6.27.1 Detailed Description

Author

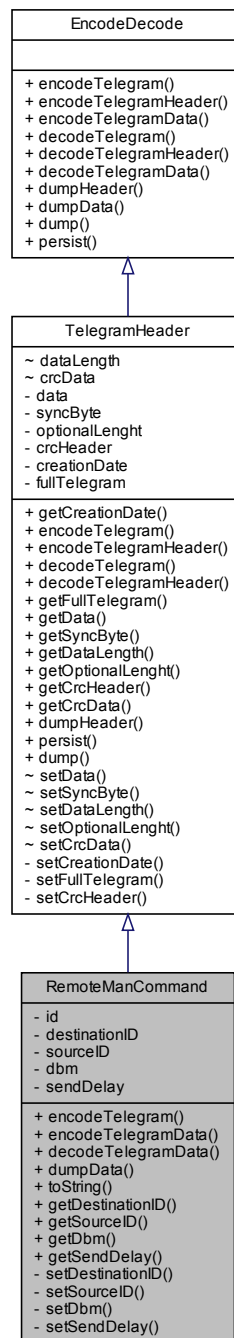
silas & louis structure. **This** (p. ??) structure is applied for the send as well as the receive case, for further informations check EnOceanSerialProtocol v3

The documentation for this class was generated from the following file:

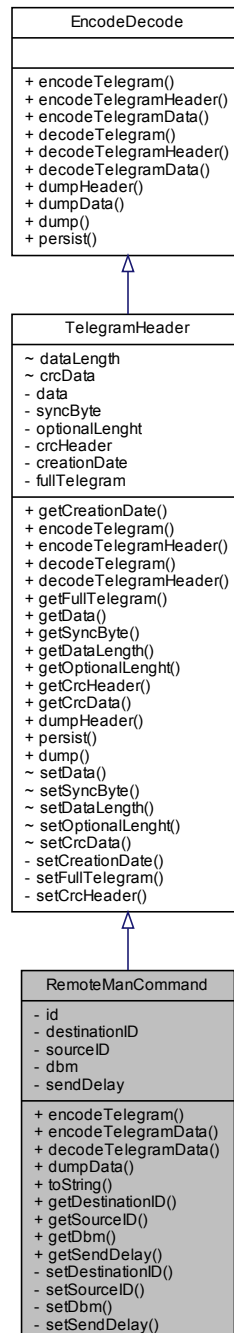
- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**RemoteManCommand**.↔
java

6.28 RemoteManCommand

Inheritance diagram for RemoteManCommand:



Collaboration diagram for RemoteManCommand:



Public Member Functions

- String **encodeTelegram** ()
encodeTelegram provides an example telegram hex for this packet type.
- String **encodeTelegramData** ()
encodeTelegramData provides an example data hex for this packet type.
- void **decodeTelegramData** (String hexTelegram)

- *decodeTelegramData splits telegram in parts*
- void **dumpData** ()
 - *dumpData Print Data, useful for logs and testiing*
- String **toString** ()
- String **getDestinationID** ()
- String **getSourceID** ()
- int **getDbm** ()
- int **getSendDelay** ()

Private Member Functions

- void **setDestinationID** (String **destinationID**)
- void **setSourceID** (String **sourceID**)
- void **setDbm** (int **dbm**)
- void **setSendDelay** (int **sendDelay**)

Private Attributes

- Long **id**
- String **destinationID**
- String **sourceID**
- int **dbm**
- int **sendDelay**

Additional Inherited Members

6.28.1 Detailed Description

Definition at line 13 of file **RemoteManCommand.java**.

6.28.2 Member Function Documentation

6.28.2.1 decodeTelegramData()

```
void decodeTelegramData (
    String hexTelegram )
```

decodeTelegramData splits telegram in parts

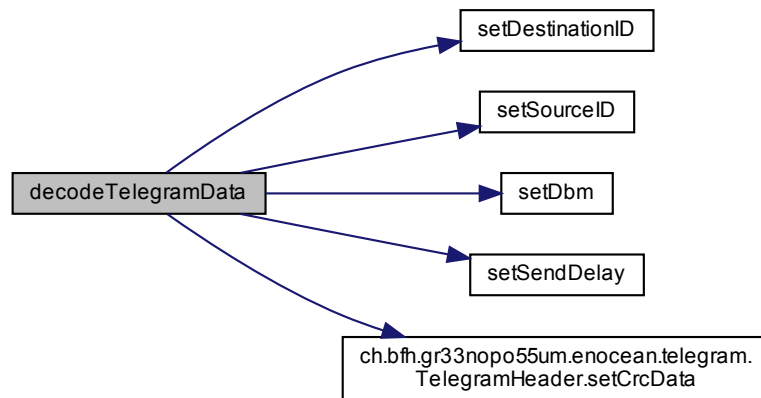
Parameters

<i>hexTelegram</i>	in hex String
--------------------	---------------

Implements **EncodeDecode** (p. ??).

Definition at line 48 of file **RemoteManCommand.java**.

Here is the call graph for this function:



6.28.2.2 dumpData()

```
void dumpData ( )
```

`dumpData` Print Data, useful for logs and testiing

Implements **EncodeDecode** (p. ??).

Definition at line **68** of file **RemoteManCommand.java**.

Here is the call graph for this function:



6.28.2.3 encodeTelegram()

```
String encodeTelegram ( )
```

`encodeTelegram` provides an example telegram hex for this packet type.

Returns

telegram hex

Implements **EncodeDecode** (p. ??).

Definition at line **30** of file **RemoteManCommand.java**.

6.28.2.4 encodeTelegramData()

```
String encodeTelegramData ( )
```

encodeTelegramData provides an example data hex for this packet type.

Returns

String hexTelegramData EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line **39** of file **RemoteManCommand.java**.

6.28.2.5 getDbm()

```
int getDbm ( )
```

Definition at line **98** of file **RemoteManCommand.java**.

6.28.2.6 getDestinationID()

```
String getDestinationID ( )
```

Definition at line **82** of file **RemoteManCommand.java**.

6.28.2.7 getSendDelay()

```
int getSendDelay ( )
```

Definition at line **106** of file **RemoteManCommand.java**.

6.28.2.8 getSourceID()

```
String getSourceID ( )
```

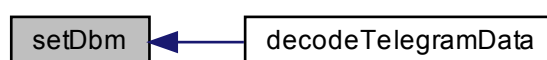
Definition at line **90** of file **RemoteManCommand.java**.

6.28.2.9 setDbm()

```
void setDbm (
    int dbm ) [private]
```

Definition at line **102** of file **RemoteManCommand.java**.

Here is the caller graph for this function:



6.28.2.10 setDestinationID()

```
void setDestinationID (  
    String destinationID ) [private]
```

Definition at line **86** of file **RemoteManCommand.java**.

Here is the caller graph for this function:



6.28.2.11 setSendDelay()

```
void setSendDelay (  
    int sendDelay ) [private]
```

Definition at line **110** of file **RemoteManCommand.java**.

Here is the caller graph for this function:

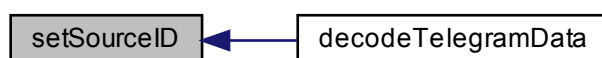


6.28.2.12 setSourceID()

```
void setSourceID (  
    String sourceID ) [private]
```

Definition at line **94** of file **RemoteManCommand.java**.

Here is the caller graph for this function:



6.28.2.13 toString()

```
String toString ( )
```

Definition at line 74 of file **RemoteManCommand.java**.

Here is the caller graph for this function:



6.28.3 Member Data Documentation

6.28.3.1 dbm

```
int dbm [private]
```

Definition at line 22 of file **RemoteManCommand.java**.

6.28.3.2 destinationID

```
String destinationID [private]
```

Definition at line 20 of file **RemoteManCommand.java**.

6.28.3.3 id

```
Long id [private]
```

Definition at line 18 of file **RemoteManCommand.java**.

6.28.3.4 sendDelay

```
int sendDelay [private]
```

Definition at line 23 of file **RemoteManCommand.java**.

6.28.3.5 sourceID

```
String sourceID [private]
```

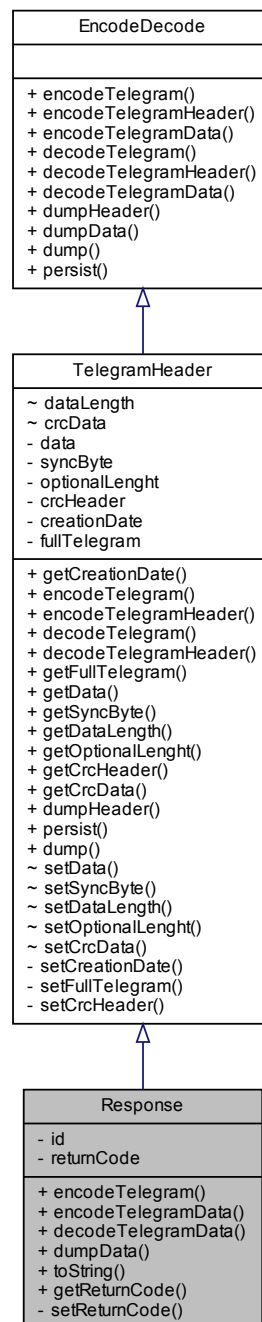
Definition at line 21 of file **RemoteManCommand.java**.

The documentation for this class was generated from the following file:

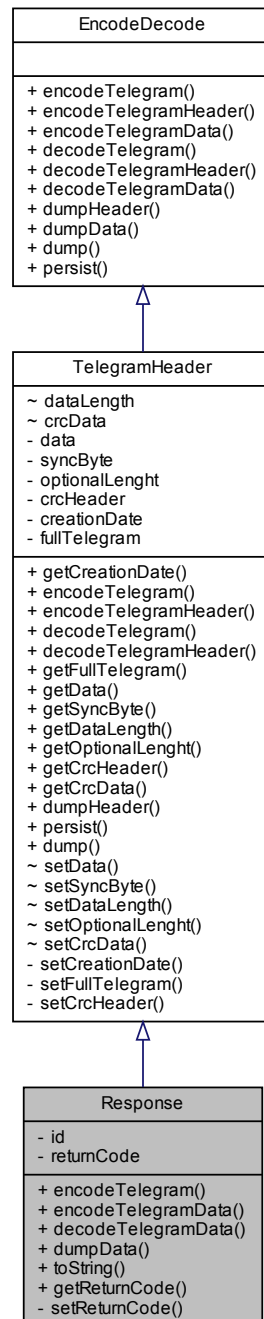
- [/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/RemoteManCommand.java](#)↔

6.29 Response

Inheritance diagram for Response:



Collaboration diagram for Response:



Public Member Functions

- String **encodeTelegram** ()
encodeTelegram provides an example telegram hex for this packet type.
- String **encodeTelegramData** ()
encodeTelegramData provides an example data hex for this packet type.
- void **decodeTelegramData** (String hexTelegram)

decodeTelegramData splits telegram in parts

- void **dumpData** ()
dumpData Print Data, useful for logs and testiing EnOcean Telegram
- String **toString** ()
- int **getReturnCode** ()

Private Member Functions

- void **setReturnCode** (int **returnCode**)

Private Attributes

- Long **id**
- int **returnCode**

Additional Inherited Members

6.29.1 Detailed Description

Definition at line 14 of file **Response.java**.

6.29.2 Member Function Documentation

6.29.2.1 decodeTelegramData()

```
void decodeTelegramData (  
    String hexTelegram )
```

decodeTelegramData splits telegram in parts

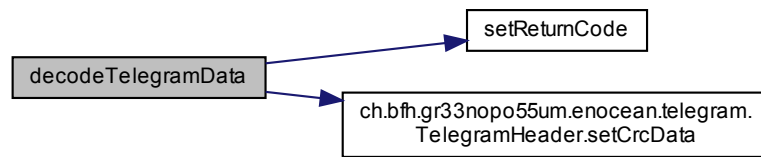
Parameters

<i>hexTelegram</i>	in hex String
--------------------	---------------

Implements **EncodeDecode** (p. ??).

Definition at line 50 of file **Response.java**.

Here is the call graph for this function:



6.29.2.2 dumpData()

```
void dumpData ( )
```

`dumpData` Print Data, useful for logs and testiing EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line **62** of file **Response.java**.

Here is the call graph for this function:



6.29.2.3 encodeTelegram()

```
String encodeTelegram ( )
```

`encodeTelegram` provides an example telegram hex for this packet type.

Returns

telegram hex

Implements **EncodeDecode** (p. ??).

Definition at line **30** of file **Response.java**.

6.29.2.4 encodeTelegramData()

```
String encodeTelegramData ( )
```

encodeTelegramData provides an example data hex for this packet type.

Returns

String hexTelegramData EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line 40 of file **Response.java**.

6.29.2.5 getReturnCode()

```
int getReturnCode ( )
```

Definition at line 73 of file **Response.java**.

6.29.2.6 setReturnCode()

```
void setReturnCode (
    int returnCode ) [private]
```

Definition at line 77 of file **Response.java**.

Here is the caller graph for this function:



6.29.2.7 toString()

```
String toString ( )
```

Definition at line 68 of file **Response.java**.

Here is the caller graph for this function:



6.29.3 Member Data Documentation

6.29.3.1 id

```
Long id [private]
```

Definition at line **19** of file **Response.java**.

6.29.3.2 returnCode

```
int returnCode [private]
```

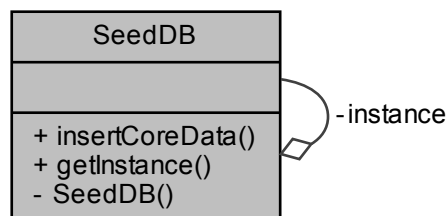
Definition at line **22** of file **Response.java**.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**Response.java**

6.30 SeedDB

Collaboration diagram for SeedDB:



Public Member Functions

- void **insertCoreData** ()
Is used to store core data in Database, e.g User, etc.

Static Public Member Functions

- static **SeedDB getInstance** ()

Private Member Functions

- **SeedDB** ()

Static Private Attributes

- static **SeedDB** instance

6.30.1 Detailed Description

Definition at line 8 of file **SeedDB.java**.

6.30.2 Constructor & Destructor Documentation

6.30.2.1 SeedDB()

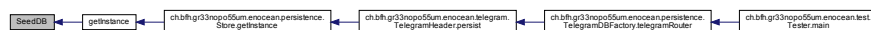
SeedDB () [private]

Definition at line 12 of file **SeedDB.java**.

Here is the call graph for this function:



Here is the caller graph for this function:



6.30.3 Member Function Documentation

6.30.3.1 getInstance()

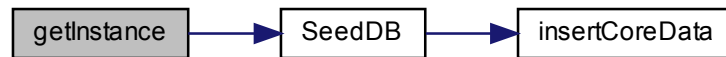
static **SeedDB** getInstance () [static]

Returns

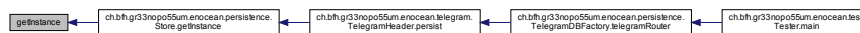
DBInstance

Definition at line **20** of file **SeedDB.java**.

Here is the call graph for this function:



Here is the caller graph for this function:

**6.30.3.2 insertCoreData()**

```
void insertCoreData ( )
```

Is used to store core data in Database, e.g User, etc.

Definition at line **30** of file **SeedDB.java**.

Here is the caller graph for this function:

**6.30.4 Member Data Documentation****6.30.4.1 instance**

```
SeedDB instance [static], [private]
```

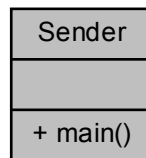
Definition at line **10** of file **SeedDB.java**.

The documentation for this class was generated from the following file:

- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nop055um/enocean/persistence/SeedDB.java`

6.31 Sender

Collaboration diagram for Sender:



Static Public Member Functions

- static void **main** (String args[])

6.31.1 Detailed Description

Used (p. ??) for Testing telegrams

Author

louis

Definition at line **14** of file **Sender.java**.

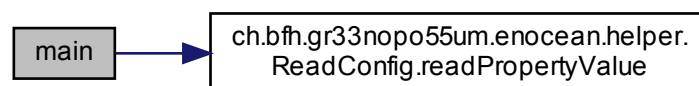
6.31.2 Member Function Documentation

6.31.2.1 main()

```
static void main (  
    String args[] ) [static]
```

Definition at line **16** of file **Sender.java**.

Here is the call graph for this function:

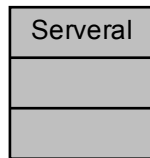


The documentation for this class was generated from the following file:

- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/test/ch/bfh/gr33nopo55um/enOcean/test/Sender.java`↔

6.32 Serveral

Collaboration diagram for Serveral:



6.32.1 Detailed Description

Author

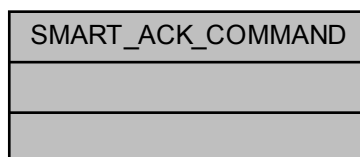
silas & louis further informations check EnOceanSerialProtocol v3

The documentation for this class was generated from the following file:

- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/CommonCommand.java`

6.33 SMART_ACK_COMMAND

Collaboration diagram for SMART_ACK_COMMAND:



6.33.1 Detailed Description

Author

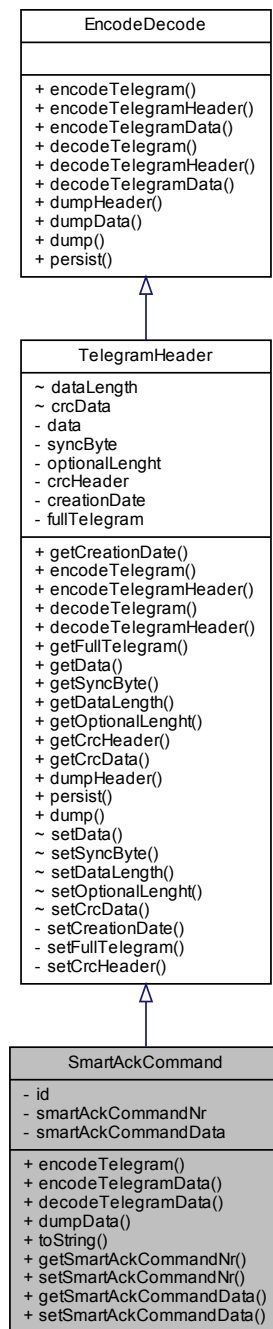
silas & louis Optional Data), for further informations check EnOceanSerialProtocol v3

The documentation for this class was generated from the following file:

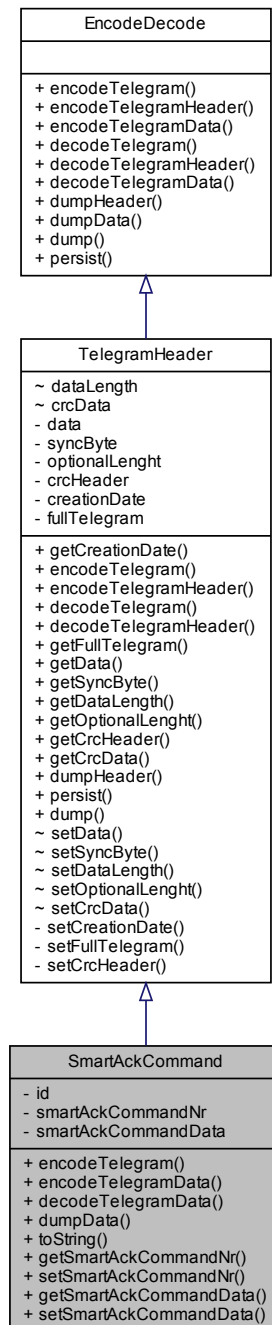
- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/SmartAckCommand.java`

6.34 SmartAckCommand

Inheritance diagram for SmartAckCommand:



Collaboration diagram for SmartAckCommand:



Public Member Functions

- String **encodeTelegram** ()
encodeTelegram provides an example telegram hex for this packet type.
- String **encodeTelegramData** ()
encodeTelegramData provides an example data hex for this packet type.
- void **decodeTelegramData** (String hexTelegram)

decodeTelegramData splits telegram in parts

- void **dumpData** ()
dumpData Print Data, useful for logs and testing
- String **toString** ()
- int **getSmartAckCommandNr** ()
- void **setSmartAckCommandNr** (int **smartAckCommandNr**)
- int **getSmartAckCommandData** ()
- void **setSmartAckCommandData** (int **smartAckCommandData**)

Private Attributes

- Long **id**
- int **smartAckCommandNr**
- int **smartAckCommandData**

Additional Inherited Members

6.34.1 Detailed Description

Definition at line 13 of file **SmartAckCommand.java**.

6.34.2 Member Function Documentation

6.34.2.1 decodeTelegramData()

```
void decodeTelegramData (  
    String hexTelegram )
```

decodeTelegramData splits telegram in parts

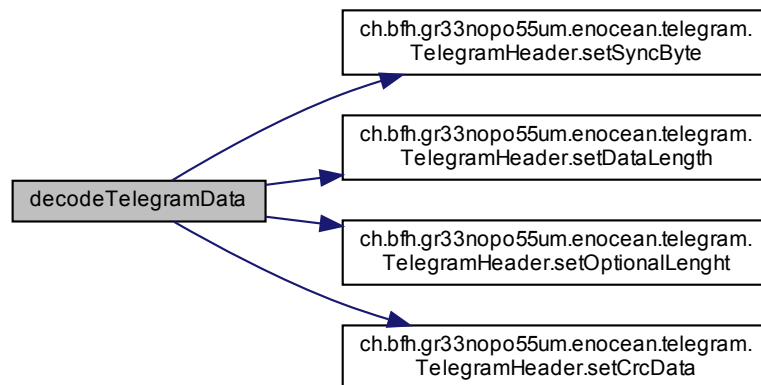
Parameters

<i>hexTelegram</i>	in hex String
--------------------	---------------

Implements **EncodeDecode** (p. ??).

Definition at line 46 of file **SmartAckCommand.java**.

Here is the call graph for this function:



6.34.2.2 dumpData()

```
void dumpData ( )
```

`dumpData` Print Data, useful for logs and testing

Implements **EncodeDecode** (p. ??).

Definition at line 61 of file **SmartAckCommand.java**.

Here is the call graph for this function:



6.34.2.3 encodeTelegram()

```
String encodeTelegram ( )
```

`encodeTelegram` provides an example telegram hex for this packet type.

Returns

telegram hex

Implements **EncodeDecode** (p. ??).

Definition at line 28 of file **SmartAckCommand.java**.

6.34.2.4 encodeTelegramData()

```
String encodeTelegramData ( )
```

encodeTelegramData provides an example data hex for this packet type.

Returns

String hexTelegramData EnOcean Telegram

Implements **EncodeDecode** (p. ??).

Definition at line **37** of file **SmartAckCommand.java**.

6.34.2.5 getSmartAckCommandData()

```
int getSmartAckCommandData ( )
```

Definition at line **80** of file **SmartAckCommand.java**.

6.34.2.6 getSmartAckCommandNr()

```
int getSmartAckCommandNr ( )
```

Definition at line **72** of file **SmartAckCommand.java**.

6.34.2.7 setSmartAckCommandData()

```
void setSmartAckCommandData (
    int smartAckCommandData )
```

Definition at line **84** of file **SmartAckCommand.java**.

6.34.2.8 setSmartAckCommandNr()

```
void setSmartAckCommandNr (
    int smartAckCommandNr )
```

Definition at line **76** of file **SmartAckCommand.java**.

6.34.2.9 toString()

```
String toString ( )
```

Definition at line **66** of file **SmartAckCommand.java**.

Here is the caller graph for this function:



6.34.3 Member Data Documentation

6.34.3.1 id

```
Long id [private]
```

Definition at line **18** of file **SmartAckCommand.java**.

6.34.3.2 smartAckCommandData

```
int smartAckCommandData [private]
```

Definition at line **21** of file **SmartAckCommand.java**.

6.34.3.3 smartAckCommandNr

```
int smartAckCommandNr [private]
```

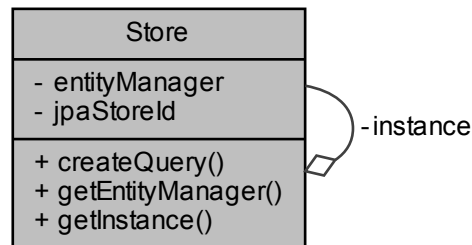
Definition at line **20** of file **SmartAckCommand.java**.

The documentation for this class was generated from the following file:

- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/SmartAckCommand.java`

6.35 Store

Collaboration diagram for Store:



Public Member Functions

- Query **createQuery** (String query)
- EntityManager **getEntityManager** ()

create a entityManager with the persistence.xml. Entity Manger is used for which class is relatet to each other...

Static Public Member Functions

- static **Store getInstance** ()

create a new Instance of a DB (p. ??)

Private Attributes

- EntityManager **entityManager** = null
- String **jpaStoreId** = "store"

Static Private Attributes

- static **Store instance**

6.35.1 Detailed Description

Definition at line 14 of file **Store.java**.

6.35.2 Member Function Documentation

6.35.2.1 createQuery()

```

Query createQuery (
    String query )
  
```

Parameters

<i>query</i>	if you want to make queries on your own. with advantage use predefined queries ;)
--------------	---

Returns

runnable query

Definition at line **39** of file **Store.java**.

6.35.2.2 getEntityManager()

```
EntityManager getEntityManager ( )
```

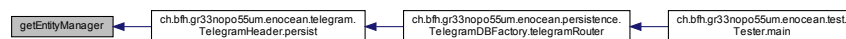
create a enitityManager with the persistence.xml. Entity Manger is used for which class is relatet to each other...

Returns

entity manager

Definition at line **48** of file **Store.java**.

Here is the caller graph for this function:

**6.35.2.3 getInstance()**

```
static Store getInstance ( ) [static]
```

create a new Instance of a **DB** (p. ??)

Returns

DB (p. ??) Instance

Definition at line **26** of file **Store.java**.

Here is the call graph for this function:



Here is the caller graph for this function:



6.35.3 Member Data Documentation

6.35.3.1 entityManager

```
EntityManager entityManager = null [private]
```

Definition at line **18** of file **Store.java**.

6.35.3.2 instance

```
Store instance [static], [private]
```

Definition at line **17** of file **Store.java**.

6.35.3.3 jpaStoreId

```
String jpaStoreId = "store" [private]
```

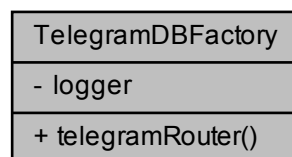
Definition at line **19** of file **Store.java**.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/persistence/**Store.java**

6.36 TelegramDBFactory

Collaboration diagram for TelegramDBFactory:



Public Member Functions

- **TelegramHeader telegramRouter** (String hexTelegram)

Static Private Attributes

- static Logger **logger** = LogManager.getLogger()

6.36.1 Detailed Description

Definition at line 11 of file **TelegramDBFactory.java**.

6.36.2 Member Function Documentation

6.36.2.1 telegramRouter()

```
TelegramHeader telegramRouter (
    String hexTelegram )
```

Parameters

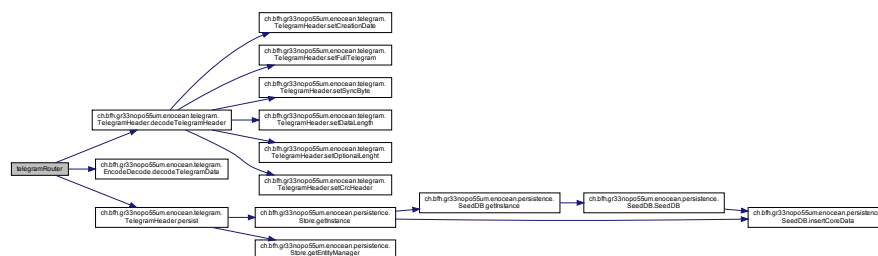
<i>hexTelegram</i>	in hex String
--------------------	---------------

Returns

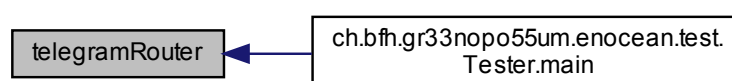
TelegramObject

Definition at line 19 of file **TelegramDBFactory.java**.

Here is the call graph for this function:



Here is the caller graph for this function:



6.36.3 Member Data Documentation

6.36.3.1 logger

```
Logger logger = LogManager.getLogger() [static], [private]
```

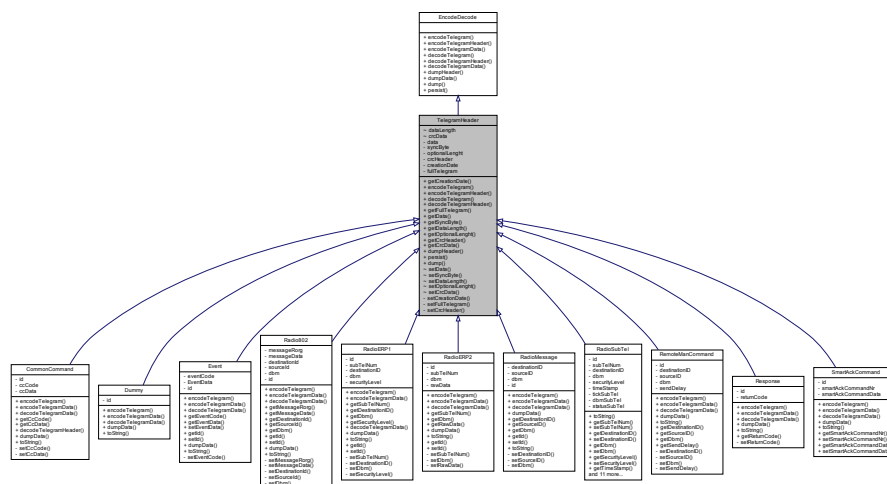
Definition at line 13 of file **TelegramDBFactory.java**.

The documentation for this class was generated from the following file:

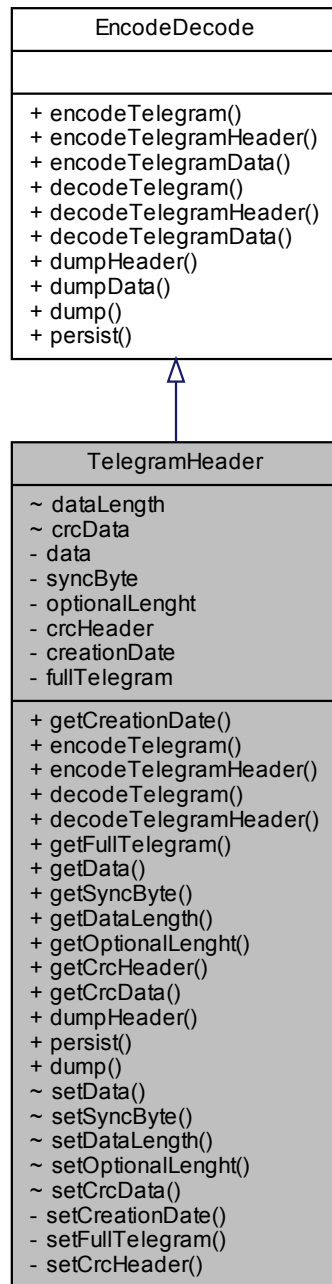
- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/persistence/**TelegramDBFactory.java**.↩

6.37 TelegramHeader

Inheritance diagram for TelegramHeader:



Collaboration diagram for TelegramHeader:



Public Member Functions

- Timestamp **getCreationDate** ()
- String **encodeTelegram** ()
- String **encodeTelegramHeader** ()
- void **decodeTelegram** (String hexTelegram)
- void **decodeTelegramHeader** (String hexTelegram)

- String **getFullTelegram** ()
- Long **getData** ()
- int **getSyncByte** ()
- int **getDataLength** ()
- int **getOptionalLenght** ()
- int **getCrcHeader** ()
- int **getCrcData** ()
- void **dumpHeader** ()
print Header in pretty print
- void **persist** ()
Create an entity manager and persist the current object.
- void **dump** ()
Used (p. ??) for debugging purposes.

Package Functions

- void **setData** (Long **data**)
- void **setSyncByte** (int **syncByte**)
- void **setDataLength** (int **dataLength**)
- void **setOptionalLenght** (int **optionalLenght**)
- void **setCrcData** (int **crcData**)

Package Attributes

- int **dataLength**
- int **crcData**

Private Member Functions

- void **setCreationDate** (Timestamp **creationDate**)
- void **setFullTelegram** (String **fullTelegram**)
- void **setCrcHeader** (int **crcHeader**)

Private Attributes

- Long **data**
- int **syncByte**
- int **optionalLenght**
- int **crcHeader**
- Timestamp **creationDate**
- String **fullTelegram**

6.37.1 Detailed Description

Author

silas & louis for all common fields of the telegrams. It also implements the Store procedure.

Definition at line **14** of file **TelegramHeader.java**.

6.37.2 Member Function Documentation

6.37.2.1 decodeTelegram()

```
void decodeTelegram (
    String hexTelegram )
```

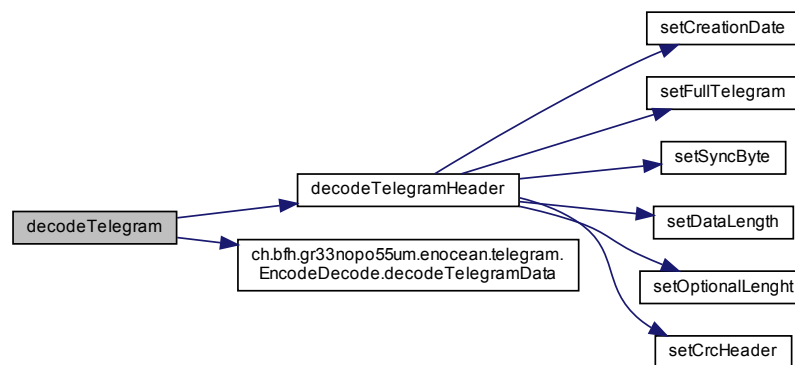
Parameters

<i>hexTelegram</i>	EnOcean Telegram
--------------------	------------------

Implements **EncodeDecode** (p. ??).

Definition at line **63** of file **TelegramHeader.java**.

Here is the call graph for this function:



6.37.2.2 decodeTelegramHeader()

```
void decodeTelegramHeader (  
    String hexTelegram )
```

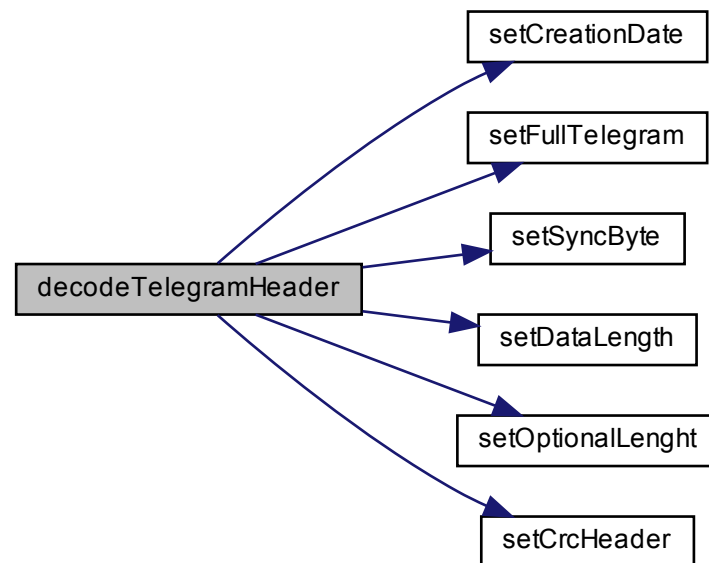
Parameters

<i>hexTelegram</i>	EnOcean Telegram
--------------------	------------------

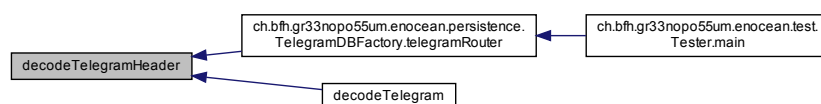
Implements **EncodeDecode** (p. ??).

Definition at line **71** of file **TelegramHeader.java**.

Here is the call graph for this function:



Here is the caller graph for this function:



6.37.2.3 dump()

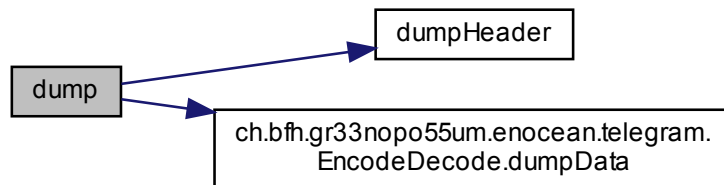
```
void dump ( )
```

Used (p. ??) for debugging purposes.

Implements **EncodeDecode** (p. ??).

Definition at line **222** of file **TelegramHeader.java**.

Here is the call graph for this function:



6.37.2.4 dumpHeader()

```
void dumpHeader ( )
```

print Header in pretty print

Implements **EncodeDecode** (p. ??).

Definition at line **192** of file **TelegramHeader.java**.

Here is the caller graph for this function:



6.37.2.5 encodeTelegram()

```
String encodeTelegram ( )
```

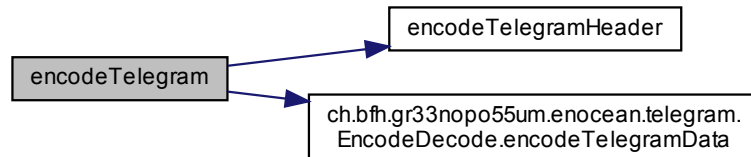
Returns

full telegram, header and data

Implements **EncodeDecode** (p. ??).

Definition at line 48 of file **TelegramHeader.java**.

Here is the call graph for this function:

**6.37.2.6 encodeTelegramHeader()**

```
String encodeTelegramHeader ( )
```

Returns

null, telegram header does not contain any data

Implements **EncodeDecode** (p. ??).

Definition at line 56 of file **TelegramHeader.java**.

Here is the caller graph for this function:

**6.37.2.7 getCrcData()**

```
int getCrcData ( )
```

Returns

data checksum

Definition at line 178 of file **TelegramHeader.java**.

6.37.2.8 getCrcHeader()

```
int getCrcHeader ( )
```

Returns

header checksum

Definition at line **162** of file **TelegramHeader.java**.

6.37.2.9 getCreationDate()

```
Timestamp getCreationDate ( )
```

Returns

creationDate

Definition at line **33** of file **TelegramHeader.java**.

6.37.2.10 getData()

```
Long getData ( )
```

Returns

Telegram data Data of the Telegram

Definition at line **101** of file **TelegramHeader.java**.

6.37.2.11 getDataLength()

```
int getDataLength ( )
```

Returns

data Length, also used to determine package length

Definition at line **131** of file **TelegramHeader.java**.

6.37.2.12 getFullTelegram()

```
String getFullTelegram ( )
```

Returns

full Telegram, helpful for finding persistence issues

Definition at line **85** of file **TelegramHeader.java**.

6.37.2.13 getOptionalLenght()

```
int getOptionalLenght ( )
```

Returns

optional Length, used for additional package information

Definition at line **146** of file **TelegramHeader.java**.

6.37.2.14 getSyncByte()

```
int getSyncByte ( )
```

Returns

syncByte is used to determine that the String recieved by Serial is an enocean Telegram

Definition at line **116** of file **TelegramHeader.java**.

6.37.2.15 persist()

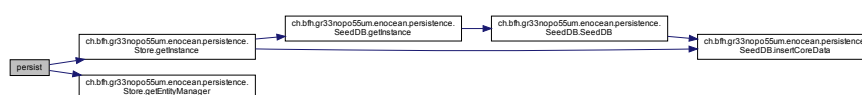
```
void persist ( )
```

Create an entity manager and persist the current object.

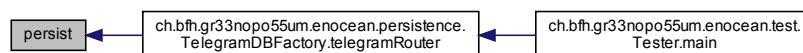
Implements **EncodeDecode** (p. ??).

Definition at line **209** of file **TelegramHeader.java**.

Here is the call graph for this function:



Here is the caller graph for this function:



6.37.2.16 setCrcData()

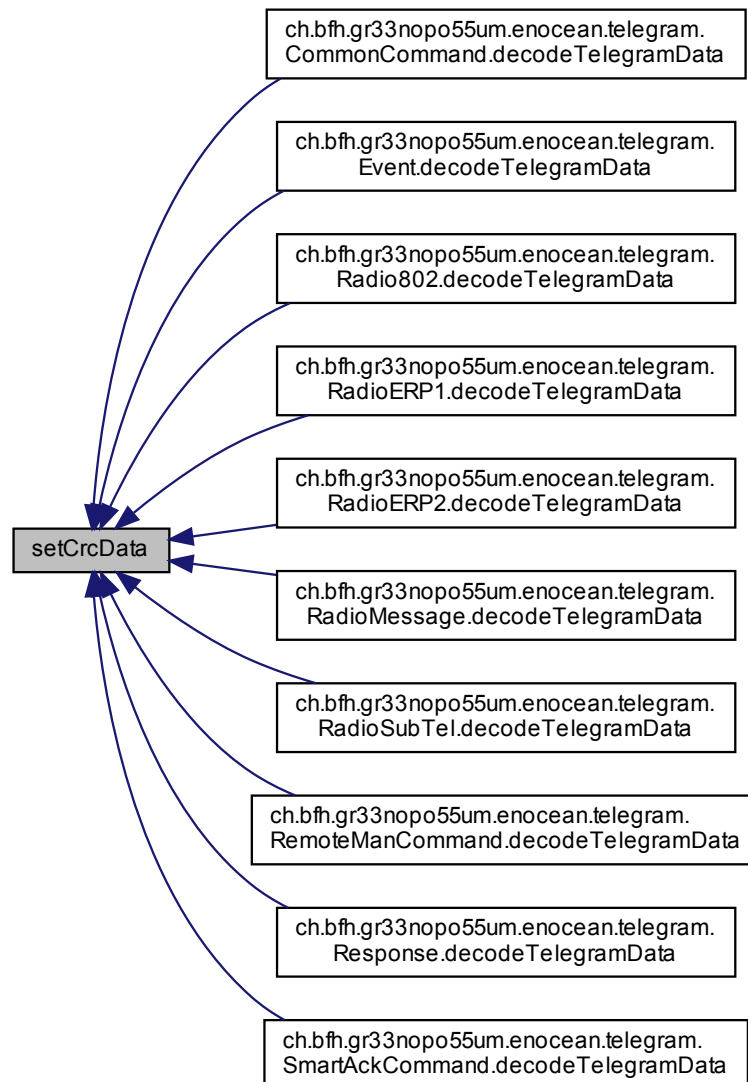
```
void setCrcData (
    int crcData ) [package]
```

Parameters

<i>crcData</i>	checksum for data
----------------	-------------------

Definition at line **185** of file **TelegramHeader.java**.

Here is the caller graph for this function:



6.37.2.17 setCrcHeader()

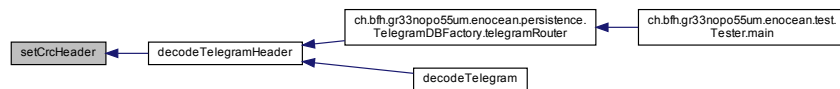
```
void setCrcHeader (  
    int crcHeader ) [private]
```


Parameters

<i>crcHeader</i>	checksum for header
------------------	---------------------

Definition at line **170** of file **TelegramHeader.java**.

Here is the caller graph for this function:



6.37.2.18 setCreationDate()

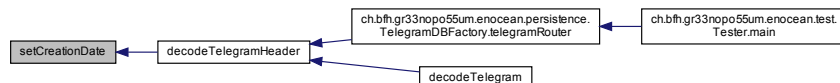
```
void setCreationDate (
    Timestamp creationDate ) [private]
```

Parameters

<i>creationDate</i>	
---------------------	--

Definition at line **41** of file **TelegramHeader.java**.

Here is the caller graph for this function:



6.37.2.19 setData()

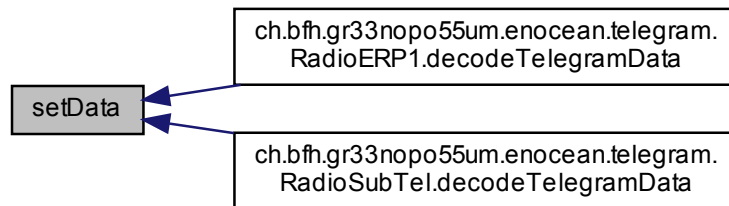
```
void setData (
    Long data ) [package]
```

Parameters

<i>data</i>	Data of the Telegram
-------------	----------------------

Definition at line **109** of file **TelegramHeader.java**.

Here is the caller graph for this function:



6.37.2.20 setDataLength()

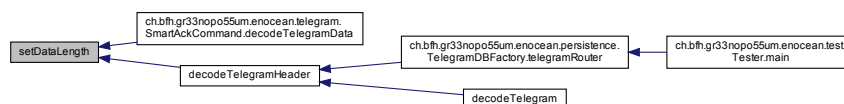
```
void setDataLength (
    int dataLength ) [package]
```

Parameters

<i>dataLength</i>	is used for mandatory data length
-------------------	-----------------------------------

Definition at line **139** of file **TelegramHeader.java**.

Here is the caller graph for this function:



6.37.2.21 setFullTelegram()

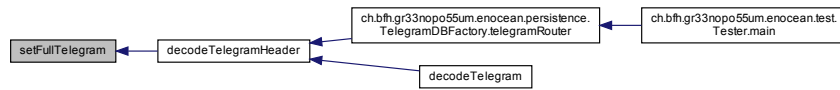
```
void setFullTelegram (
    String fullTelegram ) [private]
```

Parameters

<i>fullTelegram</i>	in hex String
---------------------	---------------

Definition at line **93** of file **TelegramHeader.java**.

Here is the caller graph for this function:



6.37.2.22 setOptionalLenght()

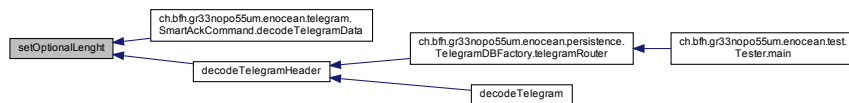
```
void setOptionalLenght (
    int optionalLenght ) [package]
```

Parameters

<i>optionalLenght</i>	is used for additional data
-----------------------	-----------------------------

Definition at line **154** of file **TelegramHeader.java**.

Here is the caller graph for this function:



6.37.2.23 setSyncByte()

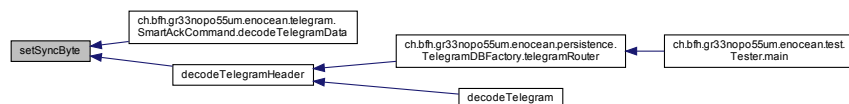
```
void setSyncByte (
    int syncByte ) [package]
```

Parameters

<i>syncByte</i>	is used to determine that the String recieved by Serial is an enocean Telegram
-----------------	--

Definition at line **123** of file **TelegramHeader.java**.

Here is the caller graph for this function:



6.37.3 Member Data Documentation

6.37.3.1 crcData

```
int crcData [package]
```

Definition at line **18** of file **TelegramHeader.java**.

6.37.3.2 crcHeader

```
int crcHeader [private]
```

Definition at line **23** of file **TelegramHeader.java**.

6.37.3.3 creationDate

```
Timestamp creationDate [private]
```

Definition at line **26** of file **TelegramHeader.java**.

6.37.3.4 data

```
Long data [private]
```

Definition at line **20** of file **TelegramHeader.java**.

6.37.3.5 dataLength

```
int dataLength [package]
```

Definition at line **17** of file **TelegramHeader.java**.

6.37.3.6 fullTelegram

```
String fullTelegram [private]
```

Definition at line **27** of file **TelegramHeader.java**.

6.37.3.7 optionalLenght

```
int optionalLenght [private]
```

Definition at line **22** of file **TelegramHeader.java**.

6.37.3.8 syncByte

```
int syncByte [private]
```

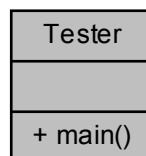
Definition at line **21** of file **TelegramHeader.java**.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/**TelegramHeader.java**

6.38 Tester

Collaboration diagram for Tester:



Static Public Member Functions

- static void **main** (String args[])

6.38.1 Detailed Description

Used (p. ??) for Testing telegrams

Author

silas

Definition at line **14** of file **Tester.java**.

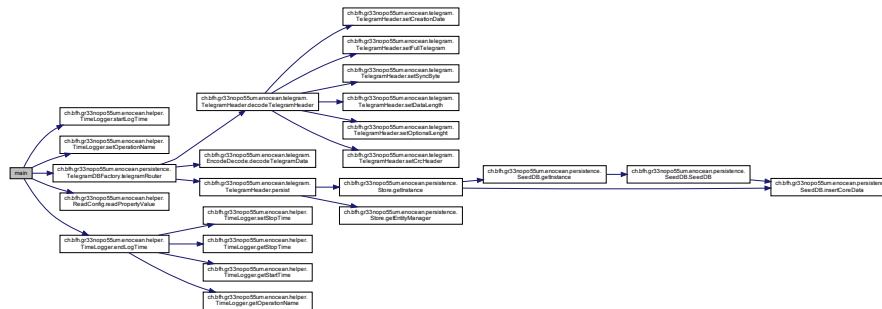
6.38.2 Member Function Documentation

6.38.2.1 main()

```
static void main (
    String args[] ) [static]
```

Definition at line 16 of file **Tester.java**.

Here is the call graph for this function:

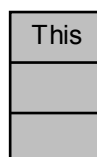


The documentation for this class was generated from the following file:

- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/test/ch/bfh/gr33nopo55um/enOcean/test/Tester.java` ↩

6.39 This

Collaboration diagram for This:



6.39.1 Detailed Description

Author

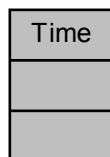
silas & louis basic encodeTelegram and decodeTelegramData methods for EnOcean telegrams.

The documentation for this class was generated from the following file:

- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/EncodeDecode.java`

6.40 Time

Collaboration diagram for Time:



6.40.1 Detailed Description

performance measurement

Author

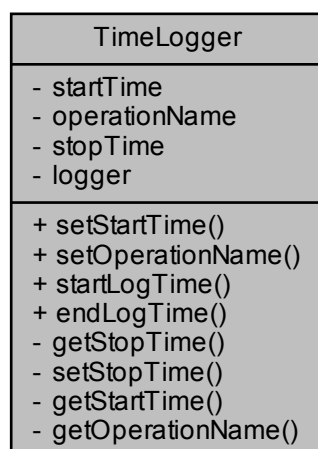
silas

The documentation for this class was generated from the following file:

- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/TimeLogger.java`

6.41 TimeLogger

Collaboration diagram for TimeLogger:



Public Member Functions

- void **setStartTime** (long **startTime**)
- void **setOperationName** (String **operationName**)
- void **startLogTime** ()
- void **endLogTime** ()

Private Member Functions

- long **getStopTime** ()
- void **setStopTime** ()
- long **getStartTime** ()
- String **getOperationName** ()

Private Attributes

- long **startTime** = 0
- String **operationName** = ""
- long **stopTime** = 0

Static Private Attributes

- static Logger **logger** = LogManager.getLogger()

6.41.1 Detailed Description

Definition at line 11 of file **TimeLogger.java**.

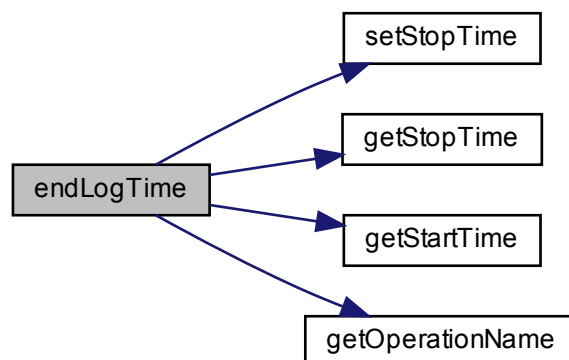
6.41.2 Member Function Documentation

6.41.2.1 endLogTime()

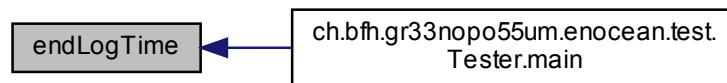
```
void endLogTime ( )
```

Definition at line 47 of file **TimeLogger.java**.

Here is the call graph for this function:



Here is the caller graph for this function:

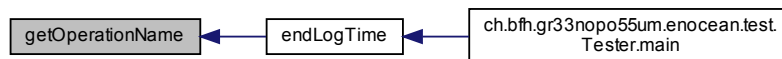


6.41.2.2 `getOperationName()`

```
String getOperationName ( ) [private]
```

Definition at line **34** of file **TimeLogger.java**.

Here is the caller graph for this function:



6.41.2.3 `getStartTime()`

```
long getStartTime ( ) [private]
```

Definition at line **26** of file **TimeLogger.java**.

Here is the caller graph for this function:

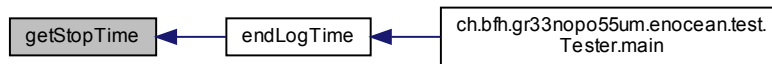


6.41.2.4 `getStopTime()`

```
long getStopTime ( ) [private]
```

Definition at line **18** of file **TimeLogger.java**.

Here is the caller graph for this function:

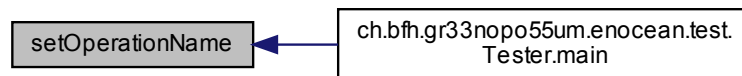


6.41.2.5 `setOperationName()`

```
void setOperationName (
    String operationName )
```

Definition at line **38** of file **TimeLogger.java**.

Here is the caller graph for this function:



6.41.2.6 `setStartTime()`

```
void setStartTime (
    long startTime )
```

Definition at line **30** of file **TimeLogger.java**.

6.41.2.7 `setStopTime()`

```
void setStopTime ( ) [private]
```

Definition at line **22** of file **TimeLogger.java**.

Here is the caller graph for this function:

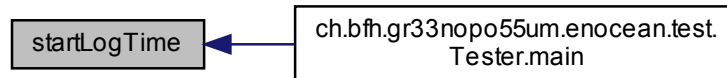


6.41.2.8 startLogTime()

```
void startLogTime ( )
```

Definition at line **42** of file **TimeLogger.java**.

Here is the caller graph for this function:



6.41.3 Member Data Documentation

6.41.3.1 logger

```
Logger logger = LogManager.getLogger() [static], [private]
```

Definition at line **13** of file **TimeLogger.java**.

6.41.3.2 operationName

```
String operationName = "" [private]
```

Definition at line **15** of file **TimeLogger.java**.

6.41.3.3 startTime

```
long startTime = 0 [private]
```

Definition at line **14** of file **TimeLogger.java**.

6.41.3.4 stopTime

```
long stopTime = 0 [private]
```

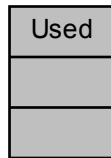
Definition at line **16** of file **TimeLogger.java**.

The documentation for this class was generated from the following file:

- `/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enocean/helper/TimeLogger.java`

6.42 Used

Collaboration diagram for Used:



6.42.1 Detailed Description

Author

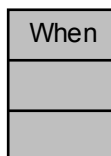
silas . Store class is implementet in telegram header.

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoclean/persistence/**Store.java**

6.43 When

Collaboration diagram for When:



6.43.1 Detailed Description

telegram, a RESPOND has to be expected. In this case, the following RESPONSE message gives the return codes: 00 RET_OK 02 RET_NOT_SUPPORTED 03 RET_WRONG_PARAM, for further informations check EnocleanSerialProtocol v3

Author

silas & louis

The documentation for this class was generated from the following file:

- /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoclean/telegram/**Response.java**

Chapter 7

File Documentation

7.1 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/application/↵ Application.java File Reference

Classes

- class **Application**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.application**

7.2 Application.java

```
00001 package ch.bfh.gr33nopo55um.enOcean.application;
00002
00007 public class Application {
00008
00009     public static void main(String args[]) throws Exception {
00010
00014         EnocleaListener enocleaListener = new EnocleaListener();
00015
00016         enocleaListener.logger();
00017     }
00018
00019 }
```

7.3 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/application/↵ EnocleaListener.java File Reference

Classes

- class **EnocleaListener**

Packages

- package **ch.bfh.gr33nopo55um.enocean.application**

7.4 EnoceListener.java

```

00001 package ch.bfh.gr33nopo55um.enocean.application;
00002
00003 import ch.bfh.gr33nopo55um.enocean.helper.ReadConfig;
00004 import it.polito.elite.enocean.enj.communication.EnJConnection;
00005 import it.polito.elite.enocean.enj.link.EnJLink;
00006 import it.polito.elite.enocean.examples.SimpleDeviceListener;
00007 import org.apache.logging.log4j.LogManager;
00008 import org.apache.logging.log4j.Logger;
00009
00010 import java.io.IOException;
00011
00016 public class EnoceListener {
00017
00018     private static Logger logger = LogManager.getLogger();
00019
00020
00024     public void logger() {
00025
00026
00027         ReadConfig readConfig = new ReadConfig();
00028         String serialPort;
00029
00033         try {
00034             serialPort = readConfig.readPropertyValue("serialPort");
00035         } catch (IOException e) {
00036             serialPort = "/dev/ttyUSB0";
00037             logger.warn("No config found, taking default Port instead: " + serialPort);
00038         }
00039
00040
00041
00046         EnJLink linkLayer = null;
00047         try {
00048             linkLayer = new EnJLink(readConfig.readPropertyValue("serialPort")
00049             );
00050         } catch (Exception e) {
00051             e.printStackTrace();
00052         }
00053
00054
00059         EnJConnection connection = new EnJConnection(linkLayer, null); //null persistent storage
00060
00061
00066         SimpleDeviceListener listener = new SimpleDeviceListener();
00067
00068
00073         connection.addEnJDeviceListener(listener);
00074
00078         linkLayer.connect();
00079
00080
00081     }
00082
00083
00084 }

```

7.5 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enocean/helper/CRC.java File Reference

Classes

- class **CRC**

Packages

- package **ch.bfh.gr33nopo55um.enocean.helper**

7.6 CRC.java

```

00001 package ch.bfh.gr33nopo55um.enocean.helper;
00002
00008 public class CRC {
00009
00010     private static byte[] crc8Table = new byte[]{
00011         (byte) 0x00, (byte) 0x07, (byte) 0x0e, (byte) 0x09, (byte) 0x1c, (byte) 0x1b, (byte) 0x12, (
00012         byte) 0x15, (byte) 0x38, (byte) 0x3f, (byte) 0x36, (byte) 0x31, (byte) 0x24, (byte) 0x23, (byte) 0x2a, (
00013         byte) 0x2d, (byte) 0x70, (byte) 0x77, (byte) 0x7e, (byte) 0x79, (byte) 0x6c, (byte) 0x6b, (byte) 0x62, (
00014         byte) 0x65, (byte) 0x48, (byte) 0x4f, (byte) 0x46, (byte) 0x41, (byte) 0x54, (byte) 0x53, (byte) 0x5a, (
00015         byte) 0x5d, (byte) 0xe0, (byte) 0xe7, (byte) 0xee, (byte) 0xe9, (byte) 0xfc, (byte) 0xfb, (byte) 0xf2, (
00016         byte) 0xf5, (byte) 0xd8, (byte) 0xdf, (byte) 0xd6, (byte) 0xd1, (byte) 0xc4, (byte) 0xc3, (byte) 0xca, (
00017         byte) 0xcd, (byte) 0x90, (byte) 0x97, (byte) 0x9e, (byte) 0x99, (byte) 0x8c, (byte) 0x8b, (byte) 0x82, (
00018         byte) 0x85, (byte) 0xa8, (byte) 0xaf, (byte) 0xa6, (byte) 0xa1, (byte) 0xb4, (byte) 0xb3, (byte) 0xba, (
00019         byte) 0xbd, (byte) 0xc7, (byte) 0xc0, (byte) 0xc9, (byte) 0xce, (byte) 0xdb, (byte) 0xdc, (byte) 0xd5, (
00020         byte) 0xd2, (byte) 0xff, (byte) 0xf8, (byte) 0xf1, (byte) 0xf6, (byte) 0xe3, (byte) 0xe4, (byte) 0xed, (
00021         byte) 0xea, (byte) 0xb7, (byte) 0xb0, (byte) 0xb9, (byte) 0xbe, (byte) 0xab, (byte) 0xac, (byte) 0xa5, (
00022         byte) 0xa2, (byte) 0x8f, (byte) 0x88, (byte) 0x81, (byte) 0x86, (byte) 0x93, (byte) 0x94, (byte) 0x9d, (
00023         byte) 0x9a, (byte) 0x27, (byte) 0x20, (byte) 0x29, (byte) 0x2e, (byte) 0x3b, (byte) 0x3c, (byte) 0x35, (
00024         byte) 0x32, (byte) 0x1f, (byte) 0x18, (byte) 0x11, (byte) 0x16, (byte) 0x03, (byte) 0x04, (byte) 0x0d, (
00025         byte) 0x0a, (byte) 0x57, (byte) 0x50, (byte) 0x59, (byte) 0x5e, (byte) 0x4b, (byte) 0x4c, (byte) 0x45, (
00026         byte) 0x42, (byte) 0x6f, (byte) 0x68, (byte) 0x61, (byte) 0x66, (byte) 0x73, (byte) 0x74, (byte) 0x7d, (
00027         byte) 0x7a, (byte) 0x89, (byte) 0x8e, (byte) 0x87, (byte) 0x80, (byte) 0x95, (byte) 0x92, (byte) 0x9b, (
00028         byte) 0x9c, (byte) 0xb1, (byte) 0xb6, (byte) 0xbf, (byte) 0xb8, (byte) 0xad, (byte) 0xaa, (byte) 0xa3, (
00029         byte) 0xa4, (byte) 0xf9, (byte) 0xfe, (byte) 0xf7, (byte) 0xf0, (byte) 0xe5, (byte) 0xe2, (byte) 0xeb, (
00030         byte) 0xec, (byte) 0xc1, (byte) 0xc6, (byte) 0xcf, (byte) 0xc8, (byte) 0xdd, (byte) 0xda, (byte) 0xd3, (
00031         byte) 0xd4, (byte) 0x69, (byte) 0x6e, (byte) 0x67, (byte) 0x60, (byte) 0x75, (byte) 0x72, (byte) 0x7b, (
00032         byte) 0x7c, (byte) 0x51, (byte) 0x56, (byte) 0x5f, (byte) 0x58, (byte) 0x4d, (byte) 0x4a, (byte) 0x43, (
00033         byte) 0x44, (byte) 0x19, (byte) 0x1e, (byte) 0x17, (byte) 0x10, (byte) 0x05, (byte) 0x02, (byte) 0x0b, (
00034         byte) 0x0c, (byte) 0x21, (byte) 0x26, (byte) 0x2f, (byte) 0x28, (byte) 0x3d, (byte) 0x3a, (byte) 0x33, (
00035         byte) 0x34, (byte) 0x4e, (byte) 0x49, (byte) 0x40, (byte) 0x47, (byte) 0x52, (byte) 0x55, (byte) 0x5c, (
00036         byte) 0x5b, (byte) 0x76, (byte) 0x71, (byte) 0x78, (byte) 0x7f, (byte) 0x6a, (byte) 0x6d, (byte) 0x64, (
00037         byte) 0x63, (byte) 0x3e, (byte) 0x39, (byte) 0x30, (byte) 0x37, (byte) 0x22, (byte) 0x25, (byte) 0x2c, (
00038         byte) 0x2b, (byte) 0x06, (byte) 0x01, (byte) 0x08, (byte) 0x0f, (byte) 0x1a, (byte) 0x1d, (byte) 0x14, (
00039         byte) 0x13, (byte) 0xae, (byte) 0xa9, (byte) 0xa0, (byte) 0xa7, (byte) 0xb2, (byte) 0xb5, (byte) 0xbc, (
00040         byte) 0xbb, (byte) 0x96, (byte) 0x91, (byte) 0x98, (byte) 0x9f, (byte) 0x8a, (byte) 0x8d, (byte) 0x84, (
00041         byte) 0x83, (byte) 0xde, (byte) 0xd9, (byte) 0xd0, (byte) 0xd7, (byte) 0xc2, (byte) 0xc5, (byte) 0xcc, (
00042         byte) 0xcb, (byte) 0xe6, (byte) 0xe1, (byte) 0xe8, (byte) 0xef, (byte) 0xfa, (byte) 0xfd, (byte) 0xf4, (
00043         byte) 0xf3});
00044
00044     public static byte calc(byte[] data) {
00045         byte crc = 0;
00046         for (byte aData : data) crc = crc8Table[(crc ^ aData) & 0xff];
00047         return crc;
00048     }

```

```
00049 }
00050
00051
```

7.7 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/ReadConfig.java File Reference ↩

Classes

- class **ReadConfig**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.helper**

7.8 ReadConfig.java

```
00001 package ch.bfh.gr33nopo55um.enOcean.helper;
00002
00003 import java.io.FileNotFoundException;
00004 import java.io.IOException;
00005 import java.io.InputStream;
00006 import java.sql.Date;
00007 import java.util.Properties;
00008
00013 public class ReadConfig {
00014     private InputStream inputStream;
00015
00016     public String readPropertyValue(String value) throws IOException {
00017         String result = null;
00018
00019         try {
00020             Properties prop = new Properties();
00021             String propFileName = "config.properties";
00022
00023             inputStream = getClass().getClassLoader().getResourceAsStream(propFileName);
00024
00025             if (inputStream != null) {
00026                 prop.load(inputStream);
00027             } else {
00028                 throw new FileNotFoundException("property file '" + propFileName + "' not found in the
classpath");
00029             }
00030
00031             Date time = new Date(System.currentTimeMillis());
00032
00037             result = prop.getProperty(value);
00038
00039
00043         } catch (Exception e) {
00044             System.out.println("Exception: " + e);
00045         } finally {
00046             assert inputStream != null;
00047             inputStream.close();
00048         }
00049         return result;
00050     }
00051 }
00052
```

7.9 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/TimeLogger.java File Reference ↩

Classes

- class **TimeLogger**

Packages

- package **ch.bfh.gr33nopo55um.enocean.helper**

7.10 TimeLogger.java

```

00001 package ch.bfh.gr33nopo55um.enocean.helper;
00002
00003 import org.apache.logging.log4j.LogManager;
00004 import org.apache.logging.log4j.Logger;
00005
00011 public class TimeLogger {
00012
00013     private static Logger logger = LogManager.getLogger();
00014     private long startTime = 0;
00015     private String operationName = "";
00016     private long stopTime = 0;
00017
00018     private long getStopTime() {
00019         return stopTime;
00020     }
00021
00022     private void setStopTime() {
00023         this.stopTime = System.currentTimeMillis();
00024     }
00025
00026     private long getStartTime() {
00027         return startTime;
00028     }
00029
00030     public void setStartTime(long startTime) {
00031         this.startTime = startTime;
00032     }
00033
00034     private String getOperationName() {
00035         return operationName;
00036     }
00037
00038     public void setOperationName(String operationName) {
00039         this.operationName = operationName;
00040     }
00041
00042     public void startLogTime() {
00043         //Log current time for performance measurement
00044         this.startTime = System.currentTimeMillis();
00045     }
00046
00047     public void endLogTime() {
00048         //Log current time for performance measurement
00049
00050         this.setStopTime();
00051
00052         long elapsedTime = this.getStopTime() - this.getStartTime();
00053         logger.info(this.getOperationName() + " in: " + elapsedTime + " milliseconds.");
00054     }
00055 }
00056 }

```

7.11 [/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enocean/persistence/SeedDB.java](#) File Reference ↩

Classes

- class **SeedDB**

Packages

- package **ch.bfh.gr33nopo55um.enocean.persistence**

7.12 SeedDB.java

```

00001 package ch.bfh.gr33nopo55um.enocean.persistence;
00002
00008 public class SeedDB {
00009
00010     private static SeedDB instance;
00011
00012     private SeedDB() {
00013         // singleton
00014         insertCoreData();
00015     }
00016
00020     public static SeedDB getInstance() {
00021         if (instance == null) {
00022             instance = new SeedDB();
00023         }
00024         return instance;
00025     }
00026
00030     public void insertCoreData() {
00031         //not neccessary yet
00032     }
00033
00034
00035 }

```

7.13 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enocean/persistence/Store.java File Reference ↩

Classes

- class **Store**

Packages

- package **ch.bfh.gr33nopo55um.enocean.persistence**

7.14 Store.java

```

00001 package ch.bfh.gr33nopo55um.enocean.persistence;
00002
00003 import javax.persistence.EntityManager;
00004 import javax.persistence.EntityManagerFactory;
00005 import javax.persistence.Persistence;
00006 import javax.persistence.Query;
00007
00008
00013 @SuppressWarnings("ALL")
00014 public class Store {
00015
00016
00017     private static Store instance;
00018     private EntityManager entityManager = null;
00019     private String jpaStoreId = "store";
00020
00021
00026     public static Store getInstance() {
00027         if (instance == null) {
00028             instance = new Store();
00029
00030             SeedDB.getInstance().insertCoreData();
00031         }
00032         return instance;
00033     }
00034
00039     public Query createQuery(String query) {

```

7.15 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/persistence/TelegramDBFactory.java File

Reference

145

```
00040     EntityManager manager = this.getEntityManager();
00041     return manager.createQuery(query);
00042 }
00043
00044 public EntityManager getEntityManager() {
00045     if (entityManager == null) {
00050         EntityManagerFactory emfactory = Persistence.createEntityManagerFactory("store");
00051         entityManager = emfactory.createEntityManager();
00052     }
00053     return entityManager;
00054 }
00055
00056
00057 }
```

7.15 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/persistence/TelegramDBFactory.java File Reference

Classes

- class **TelegramDBFactory**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.persistence**

7.16 TelegramDBFactory.java

```
00001 package ch.bfh.gr33nopo55um.enOcean.persistence;
00002
00003 import ch.bfh.gr33nopo55um.enOcean.telegram.*;
00004 import org.apache.logging.log4j.LogManager;
00005 import org.apache.logging.log4j.Logger;
00006
00011 public class TelegramDBFactory {
00012
00013     private static Logger logger = LogManager.getLogger();
00014
00019     public TelegramHeader telegramRouter(String hexTelegram) {
00020
00021         TelegramHeader telegram;
00042
00043         if (Integer.parseInt(hexTelegram.substring(2, 4), 16) == 85) {
00048
00049             switch
00051                 (Integer.parseInt(hexTelegram.substring(10, 12), 16)) {
00052
00053                 case 1:
00055                     telegram = new RadioERP1();
00056
00057                     break;
00058
00059                 case 2:
00060                     telegram = new Response();
00061                     break;
00062
00063                 case 3:
00064                     telegram = new Radio802();
00065                     break;
00066
00067                 case 4:
00068                     telegram = new Event();
00069                     break;
00070
00071                 case 5:
00072                     telegram = new CommonCommand();
00073             }
00074         }
00075     }
00076 }
```

```

00073
00074         break;
00075     case 6:
00076         telegram = new SmartAckCommand();
00077         break;
00078
00079     case 7:
00080         telegram = new SmartAckCommand();
00081
00082         break;
00083     case 9:
00084         telegram = new Radio802();
00085
00086         break;
00087     case 10:
00088         telegram = new RadioERP2();
00089
00090         break;
00091
00092     default:
00093         telegram = new Dummy();
00094         break;
00095     }
00096
00097     telegram.decodeTelegramHeader(hexTelegram);
00098     telegram.decodeTelegramData(hexTelegram);
00099     telegram.persist();
00100
00101     logger.info("EnOcean Telegram: HEX=" + hexTelegram + " " + telegram.toString());
00102
00103     return telegram;
00104 } else {
00110     logger.warn("Wrong Telegram Sync byte! Taking dummy instead");
00111     telegram = new Dummy();
00112
00113     return telegram;
00114 }
00115 }
00116 }

```

7.17 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ ↵ CommonCommand.java File Reference

Classes

- class **CommonCommand**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.telegram**

7.18 CommonCommand.java

```

00001 package ch.bfh.gr33nopo55um.enOcean.telegram;
00002
00003 import javax.persistence.Entity;
00004 import javax.persistence.GeneratedValue;
00005 import javax.persistence.GenerationType;
00006 import javax.persistence.Id;
00007
00013 @Entity
00014 public class CommonCommand extends TelegramHeader {
00015
00016
00017     @Id
00018     @GeneratedValue(strategy = GenerationType.AUTO)
00019     private Long id;
00020
00021

```

```
00022     private int ccCode;
00023     private int ccData;
00024
00025     @Override
00026     public String encodeTelegram() {
00027         return null;
00028     }
00029
00030     @Override
00031     public String encodeTelegramData() {
00032         return null;
00033     }
00034
00035     @Override
00036     public void decodeTelegramData(String hexTelegram) {
00037
00038         this.setCcCode(Integer.parseInt(hexTelegram.substring(14, 16), 16));
00039         this.setCcData(Integer.parseInt(hexTelegram.substring(16, 24), 16));
00040         this.setCrcData(crcData = Integer.parseInt(hexTelegram.substring(24, 26), 16));
00041     }
00042
00043     public int getCcCode() {
00044         return ccCode;
00045     }
00046
00047     private void setCcCode(int ccCode) {
00048         this.ccCode = ccCode;
00049     }
00050
00051     public int getCcData() {
00052         return ccData;
00053     }
00054
00055     private void setCcData(int ccData) {
00056         this.ccData = ccData;
00057     }
00058
00059     @Override
00060     public void decodeTelegramHeader(String hexTelegram) {
00061
00062     }
00063
00064     @Override
00065     public void dumpData() {
00066         System.out.println(this.toString());
00067     }
00068
00069     @Override
00070     public String toString() {
00071         return "type=CommonCommand" +
00072             " CommonCommandCode=" + ccCode +
00073             " CommonCommandData=" + ccData;
00074     }
00075 }
```

7.19 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoccean/telegram/Dummy.java File Reference ↩

Classes

- class **Dummy**

Packages

- package **ch.bfh.gr33nopo55um.enoccean.telegram**

7.20 Dummy.java

```

00001 package ch.bfh.gr33nopo55um.enocean.telegram;
00002
00003 import javax.persistence.Entity;
00004 import javax.persistence.GeneratedValue;
00005 import javax.persistence.GenerationType;
00006 import javax.persistence.Id;
00007
00012 @Entity
00013 public class Dummy extends TelegramHeader {
00014
00015     @Id
00016     @GeneratedValue(strategy = GenerationType.AUTO)
00017     private Long id;
00018
00019
00020     @Override
00021     public String encodeTelegram() {
00022         return null;
00023     }
00024
00030     @Override
00031     public String encodeTelegramData() {
00032         return null;
00033     }
00034
00035
00039     @Override
00040     public void decodeTelegramData(String hexTelegram) {
00041
00042     }
00043
00047     @Override
00048     public void dumpData() {
00049     }
00050
00051
00055     @Override
00056     public String toString() {
00057         return "PacketType=Dummy";
00058     }
00059 }

```

7.21 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enocean/telegram/↵ EncodeDecode.java File Reference

Classes

- interface **EncodeDecode**

Packages

- package **ch.bfh.gr33nopo55um.enocean.telegram**

7.22 EncodeDecode.java

```

00001 package ch.bfh.gr33nopo55um.enocean.telegram;
00002
00007 interface EncodeDecode {
00008
00013     String encodeTelegram();
00014
00019     String encodeTelegramHeader();
00020
00025     String encodeTelegramData();
00026

```

```
00031     void decodeTelegram(String hexTelegram);
00032
00037     void decodeTelegramHeader(String hexTelegram);
00038
00043     void decodeTelegramData(String hexTelegram);
00044
00048     void dumpHeader();
00049
00053     void dumpData();
00054
00058     void dump();
00059
00063     void persist();
00064
00065 }
```

7.23 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/Event.java File Reference ↩

Classes

- class **Event**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.telegram**

7.24 Event.java

```
00001 package ch.bfh.gr33nopo55um.enOcean.telegram;
00002
00003 import javax.persistence.Entity;
00004 import javax.persistence.GeneratedValue;
00005 import javax.persistence.GenerationType;
00006 import javax.persistence.Id;
00007
00012 @Entity
00013 public class Event extends TelegramHeader {
00014
00015     private int eventCode;
00016     private int eventData;
00017
00018     @Id
00019     @GeneratedValue(strategy = GenerationType.AUTO)
00020     private Long id;
00021
00022
00027     @Override
00028     public String encodeTelegram() {
00029         return null;
00030     }
00031
00036     @Override
00037     public String encodeTelegramData() {
00038         return null;
00039     }
00040
00045     @Override
00046     public void decodeTelegramData(String hexTelegram) {
00047
00048         this.setEventCode(Integer.parseInt(hexTelegram.substring(14, 16), 16));
00049         this.setCrcData(Integer.parseInt(hexTelegram.substring(16, 18), 16));
00050
00051     }
00052
00053
00054
00055     public int getEventCode() {
00056         return eventCode;
00057     }
00058 }
```

```

00057     }
00058
00059     private void setEventCode(int eventCode) {
00060         this.eventCode = eventCode;
00061     }
00062
00063     public int getEventData() {
00064         return EventData;
00065     }
00066
00067     public void setEventData(int eventData) {
00068         EventData = eventData;
00069     }
00070
00071     public Long getId() {
00072         return id;
00073     }
00074
00075     public void setId(Long id) {
00076         this.id = id;
00077     }
00078
00082     @Override
00083     public void dumpData() {
00084         System.out.println(this.toString());
00085     }
00086
00087
00088     @Override
00089     public String toString() {
00090         return "PacketType=Event" +
00091             " EventCode=" + eventCode +
00092             " EventData=" + EventData;
00093     }
00094 }

```

7.25 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/ ↩ Radio802.java File Reference

Classes

- class **Radio802**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.telegram**

7.26 Radio802.java

```

00001 package ch.bfh.gr33nopo55um.enOcean.telegram;
00002
00003 import javax.persistence.Entity;
00004 import javax.persistence.GeneratedValue;
00005 import javax.persistence.GenerationType;
00006 import javax.persistence.Id;
00007
00012 @Entity
00013 public class Radio802 extends TelegramHeader {
00014
00015     private int messageRorg;
00016     private int messageData;
00017
00018     private int destinationId;
00019
00020     private int sourceId;
00021     private int dbm;
00022     @Id
00023     @GeneratedValue(strategy = GenerationType.AUTO)
00024     private Long id;

```



```

00025
00026
00031     @Override
00032     public String encodeTelegram() {
00033         return null;
00034     }
00035
00040     @Override
00041     public String encodeTelegramData() {
00042         return null;
00043     }
00044
00049     @Override
00050     public void decodeTelegramData(String hexTelegram) {
00051
00052         this.setMessageRorg(Integer.parseInt(hexTelegram.substring(14, 16), 16));
00053         this.setMessageData(Integer.parseInt(hexTelegram.substring(16, 16 +
dataLength), 16));
00054         this.setDestinationId(Integer.parseInt(hexTelegram.substring(dataLength + 16,
dataLength + 24), 16));
00055         this.setSourceId(Integer.parseInt(hexTelegram.substring(dataLength + 24,
dataLength + 32), 16));
00056         this.setDbm(Integer.parseInt(hexTelegram.substring(dataLength + 32,
dataLength + 34), 16));
00057         this.setCrcData(Integer.parseInt(hexTelegram.substring(dataLength + 34,
dataLength + 36), 16));
00058
00059     }
00060
00061
00062     public int getMessageRorg() {
00063         return messageRorg;
00064     }
00065
00066     private void setMessageRorg(int messageRorg) {
00067         this.messageRorg = messageRorg;
00068     }
00069
00070     public int getMessageData() {
00071         return messageData;
00072     }
00073
00074     private void setMessageData(int messageData) {
00075         this.messageData = messageData;
00076     }
00077
00078     public int getDestinationId() {
00079         return destinationId;
00080     }
00081
00082     private void setDestinationId(int destinationId) {
00083         this.destinationId = destinationId;
00084     }
00085
00086     public int getSourceId() {
00087         return sourceId;
00088     }
00089
00090     private void setSourceId(int sourceId) {
00091         this.sourceId = sourceId;
00092     }
00093
00094     public int getDbm() {
00095         return dbm;
00096     }
00097
00098     private void setDbm(int dbm) {
00099         this.dbm = dbm;
00100     }
00101
00102     public Long getId() {
00103         return id;
00104     }
00105
00106     public void setId(Long id) {
00107         this.id = id;
00108     }
00109
00113     @Override
00114     public void dumpData() {
00115         System.out.println(this.toString());
00116
00117     }
00118
00119
00120     /*
00121     @brief pretty print

```

```

00122     */
00123     @Override
00124     public String toString() {
00125         return "PacketType=Radio802" +
00126             " messageRorg=" + messageRorg +
00127             " messageData=" + messageData +
00128             " destinationId=" + destinationId +
00129             " sourceId=" + sourceId +
00130             " dbm=" + dbm;
00131     }
00132 }

```

7.27 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↔ RadioERP1.java File Reference

Classes

- class **RadioERP1**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.telegram**

7.28 RadioERP1.java

```

00001 package ch.bfh.gr33nopo55um.enOcean.telegram;
00002
00003
00004 import javax.persistence.Entity;
00005 import javax.persistence.GeneratedValue;
00006 import javax.persistence.GenerationType;
00007 import javax.persistence.Id;
00008
00013 @Entity
00014 public class RadioERP1 extends TelegramHeader {
00015
00016     @Id
00017     @GeneratedValue(strategy = GenerationType.AUTO)
00018     private Long id;
00019
00020     private int subTelNum;
00021     private String destinationID;
00022     private int dbm;
00023     private int securityLevel;
00024
00025
00030     @Override
00031     public String encodeTelegram() {
00032         return null;
00033     }
00034
00039     @Override
00040     public String encodeTelegramData() {
00041         return null;
00042     }
00043
00044     public int getSubTelNum() {
00045         return subTelNum;
00046     }
00047
00048     private void setSubTelNum(int subTelNum) {
00049         this.subTelNum = subTelNum;
00050     }
00051
00052     public String getDestinationID() {
00053         return destinationID;
00054     }
00055
00056     private void setDestinationID(String destinationID) {

```

```

00057         this.destinationID = destinationID;
00058     }
00059
00060     public int getDbm() {
00061         return dbm;
00062     }
00063
00064     private void setDbm(int dbm) {
00065         this.dbm = dbm;
00066     }
00067
00068     public int getSecurityLevel() {
00069         return securityLevel;
00070     }
00071
00072     private void setSecurityLevel(int securityLevel) {
00073         this.securityLevel = securityLevel;
00074     }
00075
00080     @Override
00081     public void decodeTelegramData(String hexTelegram) {
00082
00083         //start of data
00084         int endOfDataLength = 14 + dataLength;
00085
00086
00087
00088         this.setData(Long.parseLong(hexTelegram.substring(14, endOfDataLength), 16));
00089         this.setSubTelNum(Integer.parseInt(hexTelegram.substring(endOfDataLength, endOfDataLength + 2), 16)
00090 );
00091         this.setDestinationID(hexTelegram.substring(endOfDataLength + 2, endOfDataLength + 10));
00092         this.setDbm(Integer.parseInt(hexTelegram.substring(endOfDataLength + 10, endOfDataLength + 12), 16)
00093 );
00094         this.setSecurityLevel(Integer.parseInt(hexTelegram.substring(endOfDataLength + 12, endOfDataLength
00095 + 14), 16));
00096         this.setCrcData(Integer.parseInt(hexTelegram.substring(endOfDataLength + 14, endOfDataLength + 16),
00097 16));
00098
00099     }
00100
00101     public void dumpData() {
00102
00103         System.out.println(this.toString());
00104     }
00105
00106     @Override
00107     public String toString() {
00108         return "PacketType=RadioERP1" +
00109             " SubTelNum=" + subTelNum +
00110             " DestinationID=" + destinationID +
00111             " dBm=" + dbm +
00112             " SecurityLevel=" + securityLevel;
00113     }
00114
00115     @Id
00116     public Long getId() {
00117         return id;
00118     }
00119
00120     public void setId(Long id) {
00121         this.id = id;
00122     }

```

7.29 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enoclean/telegram/↵ RadioERP2.java File Reference

Classes

- class **RadioERP2**

Packages

- package **ch.bfh.gr33nopo55um.enoclean.telegram**

7.30 RadioERP2.java

```

00001 package ch.bfh.gr33nopo55um.enocean.telegram;
00002
00003 import javax.persistence.Entity;
00004 import javax.persistence.GeneratedValue;
00005 import javax.persistence.GenerationType;
00006 import javax.persistence.Id;
00007
00013 @Entity
00014 public class RadioERP2 extends TelegramHeader {
00015
00016     @Id
00017     @GeneratedValue(strategy = GenerationType.AUTO)
00018     private Long id;
00019
00020
00021
00022     private int subTelNum;
00023     private int dbm;
00024     private String rawData;
00025
00026
00032     @Override
00033     public String encodeTelegram() {
00034         return null;
00035     }
00036
00042     @Override
00043     public String encodeTelegramData() {
00044         return null;
00045     }
00046
00052     @Override
00053     public void decodeTelegramData(String hexTelegram) {
00054
00055         this.setRawData(hexTelegram.substring(14, 26));
00056         this.setSubTelNum(Integer.parseInt(hexTelegram.substring(26, 28), 16));
00057         this.setDbm(Integer.parseInt(hexTelegram.substring(28, 30), 16));
00058         this.setCrcData(Integer.parseInt(hexTelegram.substring(30, 32), 16));
00059     }
00060
00061
00066     public int getSubTelNum() {
00067         return subTelNum;
00068     }
00069
00070     private void setSubTelNum(int subTelNum) {
00071         this.subTelNum = subTelNum;
00072     }
00073
00074     public int getDbm() {
00075         return dbm;
00076     }
00077
00078     private void setDbm(int dbm) {
00079         this.dbm = dbm;
00080     }
00081
00082     public String getRawData() {
00083         return rawData;
00084     }
00085
00086     private void setRawData(String rawData) {
00087         this.rawData = rawData;
00088     }
00089
00090
00091     @Override
00092     public void dumpData() {
00093         System.out.println(this.toString());
00094
00095     }
00096
00097
00098     @Override
00099     public String toString() {
00100         return "PacketType=RadioERP2" +
00101             " SubTelNum=" + subTelNum +
00102             " dBm=" + dbm +
00103             " RawData=" + rawData;
00104     }
00105
00106     public Long getId() {
00107         return id;
00108     }

```

```
00109
00110     public void setId(Long id) {
00111         this.id = id;
00112     }
00113 }
```

7.31 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/RadioMessage.java File Reference ↩

Classes

- class **RadioMessage**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.telegram**

7.32 RadioMessage.java

```
00001 package ch.bfh.gr33nopo55um.enOcean.telegram;
00002
00003 import javax.persistence.Entity;
00004 import javax.persistence.GeneratedValue;
00005 import javax.persistence.GenerationType;
00006 import javax.persistence.Id;
00007
00013 @Entity
00014 public class RadioMessage extends TelegramHeader {
00015     private String destinationID;
00016     private String sourceID;
00017     private int dbm;
00018
00019     @Id
00020     @GeneratedValue(strategy = GenerationType.AUTO)
00021     private Long id;
00022
00028     @Override
00029     public String encodeTelegram() {
00030         return null;
00031     }
00032
00038     @Override
00039     public String encodeTelegramData() {
00040         return null;
00041     }
00042
00048     @Override
00049     public void decodeTelegramData(String hexTelegram) {
00050
00051         this.setDestinationID(hexTelegram.substring(14, 22));
00052         this.setSourceID(hexTelegram.substring(22, 30));
00053         this.setDbm(Integer.parseInt(hexTelegram.substring(30, 32), 16));
00054         this.setCrcData(Integer.parseInt(hexTelegram.substring(32, 34), 16));
00055     }
00056
00062     @Override
00063     public void dumpData() {
00064         System.out.println(this.toString()
00065             );
00066     }
00067
00068     public String getDestinationID() {
00069         return destinationID;
00070     }
00071
00072     private void setDestinationID(String destinationID) {
00073         this.destinationID = destinationID;
00074     }
00075 }
```

```

00076     public String getSourceID() {
00077         return sourceID;
00078     }
00079
00080     private void setSourceID(String sourceID) {
00081         this.sourceID = sourceID;
00082     }
00083
00084     public int getDbm() {
00085         return dbm;
00086     }
00087
00088     private void setDbm(int dbm) {
00089         this.dbm = dbm;
00090     }
00091
00092     public Long getId() {
00093         return id;
00094     }
00095
00096     public void setId(Long id) {
00097         this.id = id;
00098     }
00099
00100     @Override
00101     public String toString() {
00102         return "PacketType=RadioMessage" +
00103             " DestinationID=" + destinationID +
00104             " SourceID=" + sourceID +
00105             " dBm=" + dbm;
00106     }
00107 }

```

7.33 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↔ RadioSubTel.java File Reference

Classes

- class **RadioSubTel**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.telegram**

7.34 RadioSubTel.java

```

00001 package ch.bfh.gr33nopo55um.enOcean.telegram;
00002
00003 import javax.persistence.Entity;
00004 import javax.persistence.GeneratedValue;
00005 import javax.persistence.GenerationType;
00006 import javax.persistence.Id;
00007
00012 @SuppressWarnings("ALL")
00013 @Entity
00014 public class RadioSubTel extends TelegramHeader {
00015
00016
00017     @Id
00018
00019     @GeneratedValue(strategy = GenerationType.AUTO)
00020     private Long id;
00021     private int subTelNum;
00022     private String destinationID;
00023     private int dbm;
00024     private int securityLevel;
00025     private String timeStamp;
00026     private String tickSubTel;
00027     private String dbmSubTel;

```

```
00028     private String statusSubTel;
00029
00030     @Override
00031     public String toString() {
00032         return "PacketType=RadioSubTel" +
00033             " SubTelNum=" + subTelNum +
00034             " DestinationID=" + destinationID +
00035             " dBm=" + dbm +
00036             " SecurityLevel=" + securityLevel +
00037             " TimeStamp=" + timeStamp +
00038             " TickSubTel=" + tickSubTel +
00039             " dBmSubTel=" + dbmSubTel +
00040             " StatusSubTel='" + statusSubTel;
00041     }
00042
00043     public int getSubTelNum() {
00044         return subTelNum;
00045     }
00046
00047     public void setSubTelNum(int subTelNum) {
00048         this.subTelNum = subTelNum;
00049     }
00050
00051     public String getDestinationID() {
00052         return destinationID;
00053     }
00054
00055     public void setDestinationID(String destinationID) {
00056         this.destinationID = destinationID;
00057     }
00058
00059     public int getDbm() {
00060         return dbm;
00061     }
00062
00063     public void setDbm(int dbm) {
00064         this.dbm = dbm;
00065     }
00066
00067     public int getSecurityLevel() {
00068         return securityLevel;
00069     }
00070
00071     public void setSecurityLevel(int securityLevel) {
00072         this.securityLevel = securityLevel;
00073     }
00074
00075     public String getTimeStamp() {
00076         return timeStamp;
00077     }
00078
00079     public void setTimeStamp(String timeStamp) {
00080         this.timeStamp = timeStamp;
00081     }
00082
00083     public String getTickSubTel() {
00084         return tickSubTel;
00085     }
00086
00087     public void setTickSubTel(String tickSubTel) {
00088         this.tickSubTel = tickSubTel;
00089     }
00090
00091     public String getDbmSubTel() {
00092         return dbmSubTel;
00093     }
00094
00095     public void setDbmSubTel(String dbmSubTel) {
00096         this.dbmSubTel = dbmSubTel;
00097     }
00098
00099     public String getStatusSubTel() {
00100         return statusSubTel;
00101     }
00102
00103     public void setStatusSubTel(String statusSubTel) {
00104         this.statusSubTel = statusSubTel;
00105     }
00106
00111     @Override
00112     public String encodeTelegram() {
00113         return null;
00114     }
00115
00120     @Override
00121     public String encodeTelegramData() {
00122         return null;
```

```

00123     }
00124
00125
00130     @Override
00131     public void decodeTelegramData(String hexTelegram) {
00132
00133
00134         this.setData(Long.parseLong(hexTelegram.substring(14, 14 + dataLength), 16));
00135
00136         this.setSubTelNum(Integer.parseInt(hexTelegram.substring(dataLength + 14,
00137 dataLength + 16), 16));
00138
00139         this.setDestinationID(hexTelegram.substring(dataLength + 16, dataLength + 24));
00140
00141         this.setDbm(Integer.parseInt(hexTelegram.substring(dataLength + 24,
00142 dataLength + 26), 16));
00143
00144         this.setSecurityLevel(securityLevel = Integer.parseInt(hexTelegram.substring(
00145 dataLength + 26, dataLength + 28), 16));
00146
00147         this.setTimeStamp(timeStamp = hexTelegram.substring(dataLength + 28,
00148 dataLength + 32));
00149
00150         this.setTickSubTel(tickSubTel = hexTelegram.substring(dataLength + 32,
00151 dataLength + 34));
00152
00153         this.setDbmSubTel(dbmSubTel = hexTelegram.substring(dataLength + 34,
00154 dataLength + 36));
00155
00156         this.setStatusSubTel(statusSubTel = hexTelegram.substring(dataLength + 36,
00157 dataLength + 38));
00158
00159         this.setCrcData(crcData = Integer.parseInt(hexTelegram.substring(
00160 dataLength + 38, dataLength + 40), 16));
00161
00162     }
00163
00164     @Override
00165     public void dumpData() {
00166         System.out.println(this.toString());
00167     }
00168 }

```

7.35 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/RemoteManCommand.java File Reference ↩

Classes

- class **RemoteManCommand**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.telegram**

7.36 RemoteManCommand.java

```

00001 package ch.bfh.gr33nopo55um.enOcean.telegram;
00002
00003 import javax.persistence.Entity;
00004 import javax.persistence.GeneratedValue;
00005 import javax.persistence.GenerationType;
00006 import javax.persistence.Id;
00007
00012 @Entity
00013 public class RemoteManCommand extends TelegramHeader {
00014
00015
00016     @Id
00017     @GeneratedValue(strategy = GenerationType.AUTO)

```



```
00018     private Long id;
00019
00020     private String destinationID;
00021     private String sourceID;
00022     private int dbm;
00023     private int sendDelay;
00024
00029     @Override
00030     public String encodeTelegram() {
00031         return null;
00032     }
00033
00038     @Override
00039     public String encodeTelegramData() {
00040         return null;
00041     }
00042
00047     @Override
00048     public void decodeTelegramData(String hexTelegram) {
00049
00050
00051         this.setDestinationID(hexTelegram.substring(14, 22));
00052
00053         this.setSourceID(hexTelegram.substring(22, 30));
00054
00055         this.setDbm(Integer.parseInt(hexTelegram.substring(30, 32), 16));
00056
00057         this.setSendDelay(Integer.parseInt(hexTelegram.substring(32, 34), 16));
00058
00059         this.setCrcData(Integer.parseInt(hexTelegram.substring(34, 36), 16));
00060
00061     }
00062
00063
00067     @Override
00068     public void dumpData() {
00069         System.out.println(this.toString());
00070     }
00071
00072
00073     @Override
00074     public String toString() {
00075         return "PacketType=RemoteManCommand" +
00076             " DestinationID=" + destinationID +
00077             " SourceID=" + sourceID +
00078             " dBm=" + dbm +
00079             " SendWithDelay=" + sendDelay;
00080     }
00081
00082     public String getDestinationID() {
00083         return destinationID;
00084     }
00085
00086     private void setDestinationID(String destinationID) {
00087         this.destinationID = destinationID;
00088     }
00089
00090     public String getSourceID() {
00091         return sourceID;
00092     }
00093
00094     private void setSourceID(String sourceID) {
00095         this.sourceID = sourceID;
00096     }
00097
00098     public int getDbm() {
00099         return dbm;
00100     }
00101
00102     private void setDbm(int dbm) {
00103         this.dbm = dbm;
00104     }
00105
00106     public int getSendDelay() {
00107         return sendDelay;
00108     }
00109
00110     private void setSendDelay(int sendDelay) {
00111         this.sendDelay = sendDelay;
00112     }
00113 }
```

7.37 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↔ Response.java File Reference

Classes

- class **Response**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.telegram**

7.38 Response.java

```

00001 package ch.bfh.gr33nopo55um.enOcean.telegram;
00002
00003 import javax.persistence.Entity;
00004 import javax.persistence.GeneratedValue;
00005 import javax.persistence.GenerationType;
00006 import javax.persistence.Id;
00007
00013 @Entity
00014 public class Response extends TelegramHeader {
00015
00016
00017     @Id
00018     @GeneratedValue(strategy = GenerationType.AUTO)
00019     private Long id;
00020
00021
00022     private int returnCode;
00023
00029     @Override
00030     public String encodeTelegram() {
00031         return null;
00032     }
00033
00039     @Override
00040     public String encodeTelegramData() {
00041         return null;
00042     }
00043
00049     @Override
00050     public void decodeTelegramData(String hexTelegram) {
00051
00052         this.setReturnCode(Integer.parseInt(hexTelegram.substring(14, 4), 16));
00053         this.setCrcData(Integer.parseInt(hexTelegram.substring(14, 16), 16));
00054
00055     }
00056
00061     @Override
00062     public void dumpData() {
00063         System.out.println(this.toString());
00064     }
00065
00066
00067     @Override
00068     public String toString() {
00069         return "PacketType=Response" +
00070             " ReturnCode=" + returnCode;
00071     }
00072
00073     public int getReturnCode() {
00074         return returnCode;
00075     }
00076
00077     private void setReturnCode(int returnCode) {
00078         this.returnCode = returnCode;
00079     }
00080 }
00081

```

SmartAckCommand.java File Reference

Classes

- class **SmartAckCommand**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.telegram**

7.40 SmartAckCommand.java

```

00001 package ch.bfh.gr33nopo55um.enOcean.telegram;
00002
00003 import javax.persistence.Entity;
00004 import javax.persistence.GeneratedValue;
00005 import javax.persistence.GenerationType;
00006 import javax.persistence.Id;
00007
00012 @Entity
00013 public class SmartAckCommand extends TelegramHeader {
00014
00015
00016     @Id
00017     @GeneratedValue(strategy = GenerationType.AUTO)
00018     private Long id;
00019
00020     private int smartAckCommandNr;
00021     private int smartAckCommandData;
00022
00027     @Override
00028     public String encodeTelegram() {
00029         return null;
00030     }
00031
00036     @Override
00037     public String encodeTelegramData() {
00038         return null;
00039     }
00040
00045     @Override
00046     public void decodeTelegramData(String hexTelegram) {
00047
00048         this.setSyncByte(Integer.parseInt(hexTelegram.substring(2, 4), 16));
00049         this.setDataLength(Integer.parseInt(hexTelegram.substring(4, 8), 16));
00050
00051         this.setOptionalLenght(Integer.parseInt(hexTelegram.substring(8, 10), 16));
00052         this.setCrcData(Integer.parseInt(hexTelegram.substring(12, 14), 16));
00053
00054
00055     }
00056
00060     @Override
00061     public void dumpData() {
00062         System.out.println(this.toString());
00063     }
00064
00065     @Override
00066     public String toString() {
00067         return "PacketType=SmartAckCommand" +
00068             " SmartAckCommandNr=" + smartAckCommandNr +
00069             " SmartAckData=" + smartAckCommandData;
00070     }
00071
00072     public int getSmartAckCommandNr() {
00073         return smartAckCommandNr;
00074     }
00075
00076     public void setSmartAckCommandNr(int smartAckCommandNr) {
00077         this.smartAckCommandNr = smartAckCommandNr;
00078     }
00079

```

```

00080     public int getSmartAckCommandData() {
00081         return smartAckCommandData;
00082     }
00083
00084     public void setSmartAckCommandData(int smartAckCommandData) {
00085         this.smartAckCommandData = smartAckCommandData;
00086     }
00087 }

```

7.41 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/TelegramHeader.java File Reference ↩

Classes

- class **TelegramHeader**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.telegram**

7.42 TelegramHeader.java

```

00001 package ch.bfh.gr33nopo55um.enOcean.telegram;
00002
00003 import ch.bfh.gr33nopo55um.enOcean.persistence.Store;
00004
00005 import javax.persistence.Column;
00006 import javax.persistence.MappedSuperclass;
00007 import java.sql.Timestamp;
00008
00013 @MappedSuperclass
00014 public abstract class TelegramHeader implements EncodeDecode {
00015
00016
00017     int dataLength;
00018     int crcData;
00019     //some data could be longer than int
00020     private Long data;
00021     private int syncByte;
00022     private int optionalLength;
00023     private int crcHeader;
00024
00025     @Column(name = "creationDate")
00026     private Timestamp creationDate;
00027     private String fullTelegram;
00028
00029
00033     public Timestamp getCreationDate() {
00034         return creationDate;
00035     }
00036
00037
00041     private void setCreationDate(Timestamp creationDate) {
00042         this.creationDate = creationDate;
00043     }
00044
00048     public String encodeTelegram() {
00049         return encodeTelegramHeader() + encodeTelegramData();
00050     }
00051
00052
00056     public String encodeTelegramHeader() {
00057         return null;
00058     }
00059
00063     public void decodeTelegram(String hexTelegram) {
00064         decodeTelegramHeader(hexTelegram);
00065         decodeTelegramData(hexTelegram);
00066     }

```

```

00067
00071 public void decodeTelegramHeader(String hexTelegram) {
00072     Timestamp timestamp = new Timestamp(System.currentTimeMillis());
00073     this.setCreationDate(timestamp);
00074     this.setFullTelegram(hexTelegram);
00075     this.setSyncByte(Integer.parseInt(hexTelegram.substring(2, 4), 16));
00076     this.setDataLength(Integer.parseInt(hexTelegram.substring(4, 8), 16));
00077     this.setOptionalLenght(Integer.parseInt(hexTelegram.substring(8, 10), 16));
00078     this.setCrcHeader(Integer.parseInt(hexTelegram.substring(12, 14), 16));
00079 }
00080
00081
00085 public String getFullTelegram() {
00086     return fullTelegram;
00087 }
00088
00093 private void setFullTelegram(String fullTelegram) {
00094     this.fullTelegram = fullTelegram;
00095 }
00096
00097
00101 public Long getData() {
00102     return data;
00103 }
00104
00105
00109 void setData(Long data) {
00110     this.data = data;
00111 }
00112
00116 public int getSyncByte() {
00117     return syncByte;
00118 }
00119
00123 void setSyncByte(int syncByte) {
00124     this.syncByte = syncByte;
00125 }
00126
00127
00131 public int getDataLength() {
00132     return dataLength;
00133 }
00134
00135
00139 void setDataLength(int dataLength) {
00140     this.dataLength = dataLength;
00141 }
00142
00146 public int getOptionalLenght() {
00147     return optionalLenght;
00148 }
00149
00150
00154 void setOptionalLenght(int optionalLenght) {
00155     this.optionalLenght = optionalLenght;
00156 }
00157
00158
00162 public int getCrcHeader() {
00163     return crcHeader;
00164 }
00165
00166
00170 private void setCrcHeader(int crcHeader) {
00171     this.crcHeader = crcHeader;
00172 }
00173
00174
00178 public int getCrcData() {
00179     return crcData;
00180 }
00181
00185 void setCrcData(int crcData) {
00186     this.crcData = crcData;
00187 }
00188
00192 public void dumpHeader() {
00193
00194     System.out.println("TelegramHeader{" +
00195         "data=" + data +
00196         ", syncByte=" + syncByte +
00197         ", dataLength=" + dataLength +
00198         ", optionalLenght=" + optionalLenght +
00199         ", crcData=" + crcData +
00200         ", crcHeader=" + crcHeader +
00201         ", creationDate=" + creationDate +
00202         ", fullTelegram='" + fullTelegram + '\'' + '}' + '}');

```

```

00203
00204     }
00205
00209     public void persist() {
00210
00211         Store.getInstance().getEntityManager().getTransaction().begin();
00212         Store.getInstance().getEntityManager().persist(this);
00213         Store.getInstance().getEntityManager().getTransaction().commit();
00214
00215
00216     }
00217
00218
00222     public void dump() {
00223
00224         System.out.println("Echo: " + this.getClass().toString() + "telegram");
00225
00226         dumpHeader();
00227         dumpData();
00228     }
00229
00230 }

```

7.43 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/test/ch/bfh/gr33nopo55um/ Sender.java File Reference

Classes

- class **Sender**

Packages

- package **ch.bfh.gr33nopo55um.enOcean.test**

7.44 Sender.java

```

00001 package ch.bfh.gr33nopo55um.enOcean.test;
00002
00003 import ch.bfh.gr33nopo55um.enOcean.helper.ReadConfig;
00004 import it.polito.elite.enOcean.enj.link.EnJLink;
00005 import it.polito.elite.enOcean.protocol.serial.v3.network.packet.ESP3Packet;
00006
00007 import java.io.IOException;
00008
00014 public class Sender {
00015
00016     public static void main(String args[]) {
00017
00018         ReadConfig readConfig = new ReadConfig();
00019         try {
00020             System.out.println(readConfig.readPropertyValue("serialPort"));
00021         } catch (IOException e) {
00022             e.printStackTrace();
00023         }
00024
00025         EnJLink linkLayer = null;
00026         try {
00027             linkLayer = new EnJLink(readConfig.readPropertyValue("serialPort")
00028                 );
00029         } catch (Exception e) {
00030             e.printStackTrace();
00031         }
00032         linkLayer.connect();
00033
00034         /*
00035         * Loop for masssending 0x55 000a 07 01 eb a50000000c019134fc00 01 ffffffff 3d 00 2f
00036         */
00037         for (int i = 0; i < 10; i++) {
00038             byte packetType = (byte)1;

```

```

00039         byte[] data = new byte[] { (byte)0xa5, (byte)0x00, (byte)0x00, (byte)0x00, (byte)0x0c, (byte)
0x01, (byte)0x91, (byte)0x34, (byte)0xfc, (byte)0x00 };
00040         byte[] optData = new byte[] { (byte)0x01, (byte)0xff, (byte)0xff, (byte)0xff, (byte)0xff, (byte)
0x3d, (byte)0x00 };
00041
00042         ESP3Packet esp3Packet = new ESP3Packet(packetType, data, optData);
00043         esp3Packet.buildPacket();
00044         linkLayer.send(esp3Packet);
00045         try {
00046             Thread.sleep(200);
00047         } catch (InterruptedException e) {
00048             e.printStackTrace();
00049         }
00050     }
00051
00052     /*
00053     * Loop for masssending 0x55 0007 07 01 7af6000022a13420 01 ffffffff 33 00 88
00054     */
00055     for (int i = 0; i < 10; i++) {
00056         byte packetType = (byte)1;
00057         byte[] data = new byte[] { (byte)0x7a, (byte)0xf6, (byte)0x00, (byte)0x00, (byte)0x22, (byte)
0xa1, (byte)0x34, (byte)0x20 };
00058         byte[] optData = new byte[] { (byte)0x01, (byte)0xff, (byte)0xff, (byte)0xff, (byte)0xff, (byte)
0x33, (byte)0x00 };
00059
00060         ESP3Packet esp3Packet = new ESP3Packet(packetType, data, optData);
00061         esp3Packet.buildPacket();
00062         linkLayer.send(esp3Packet);
00063         try {
00064             Thread.sleep(200);
00065         } catch (InterruptedException e) {
00066             e.printStackTrace();
00067         }
00068     }
00069 }
00070 }

```

7.45 /home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enocan/test/ch/bfh/gr33nopo55um/enocan/test/Tester.java File Reference

Classes

- class **Tester**

Packages

- package **ch.bfh.gr33nopo55um.enocan.test**

7.46 Tester.java

```

00001 package ch.bfh.gr33nopo55um.enocan.test;
00002
00003 import ch.bfh.gr33nopo55um.enocan.helper.ReadConfig;
00004 import ch.bfh.gr33nopo55um.enocan.helper.TimeLogger;
00005 import ch.bfh.gr33nopo55um.enocan.persistence.TelegramDBFactory;
00006
00007 import java.io.IOException;
00008
00014 public class Tester {
00015
00016     public static void main(String args[]) {
00017
00018
00019         TimeLogger timeLogger = new TimeLogger();
00020
00021         timeLogger.startLogTime();
00022         timeLogger.setOperationName("Application");
00023
00024         TelegramDBFactory telegramDBFactory = new TelegramDBFactory();

```

```
00025
00026      /*
00027      * @brief Loop for masstesting
00028      */
00029      for (int i = 0; i < 20; i++) {
00030          telegramDBFactory.telegramRouter("0x55000707017af6000022a1342001ffffffff3a0035");
00031          telegramDBFactory.telegramRouter("0x55000a0701eba50000000c019134fc0001ffffffff3d002f");
00032      }
00033      ReadConfig readConfig = new ReadConfig();
00034      try {
00035          System.out.println(readConfig.readPropertyValue("serialPort"));
00036      } catch (IOException e) {
00037          e.printStackTrace();
00038      }
00039
00040
00041      timeLogger.endLogTime();
00042  }
00043
00044
00045 }
```


Index

/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/application/↵
Application.java, 139 calc
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/application/↵
EnocealListener.java, 139, 140 ch.bfh.gr33nopo55um::enocean::helper::CRC, 21
CRC.java, 140, 141 ccCode
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/↵
CRC.java, 140, 141 ch.bfh.gr33nopo55um::enocean::telegram::↵
CommonCommand, 20
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/↵
ReadConfig.java, 142 ch.bfh.gr33nopo55um::enocean::telegram::↵
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/helper/↵
CommonCommand, 20
TimeLogger.java, 142, 143 ch, 9
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/persistence/↵
SeedDB.java, 143, 144 ch.bfh.gr33nopo55um, 9
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/persistence/↵
Store.java, 144 ch.bfh.gr33nopo55um::enocean, 9
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/persistence/↵
TelegramDBFactory.java, 145 ch.bfh.gr33nopo55um.enOcean.application, 10
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/persistence/↵
TelegramDBFactory.java, 145 ch.bfh.gr33nopo55um::enocean.helper, 10
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
CommonCommand.java, 146 ch.bfh.gr33nopo55um.enOcean.persistence, 10
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
CommonCommand.java, 146 ch.bfh.gr33nopo55um::enocean.telegram, 10
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
Dummy.java, 147, 148 ch.bfh.gr33nopo55um::enocean.test, 10
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
Application
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
main, 13
EncodeDecode.java, 148 ch.bfh.gr33nopo55um::enocean::application::↵
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
EnocealListener
Event.java, 149 logger, 33
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
Radio802.java, 150 ch.bfh.gr33nopo55um::enocean::helper::CRC
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
RadioERP1.java, 152 calc, 21
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
RadioERP2.java, 153, 154 crc8Table, 22
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
RadioERP2.java, 153, 154 ch.bfh.gr33nopo55um::enocean::helper::ReadConfig
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
RadioMessage.java, 155 inputStream, 88
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
RadioMessage.java, 155 readPropertyValue, 87
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
RadioSubTel.java, 156 ch.bfh.gr33nopo55um::enocean::helper::TimeLogger
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
RadioSubTel.java, 156 endLogTime, 134
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
RemoteManCommand.java, 158 getOperationName, 135
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
RemoteManCommand.java, 158 getStartTime, 135
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
Response.java, 160 getStopTime, 135
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
Response.java, 160 logger, 137
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
SmartAckCommand.java, 161 operationName, 137
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
SmartAckCommand.java, 161 setOperationName, 136
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
TelegramHeader.java, 162 setStartTime, 136
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
TelegramHeader.java, 162 setStopTime, 136
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/telegram/↵
TelegramHeader.java, 162 startLogTime, 136
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/test/ch/bfh/gr33nopo55um/enOcean/test/↵
Sender.java, 164 startTime, 137
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/test/ch/bfh/gr33nopo55um/enOcean/test/↵
Tester.java, 165 stopTime, 137
/home/silas/IdeaProjects/EnOcean/src/ch/bfh/gr33nopo55um/enOcean/test/ch/bfh/gr33nopo55um/enOcean/test/↵
Tester.java, 165 ch.bfh.gr33nopo55um::enocean.persistence::SeedDB
getInstance, 103
insertCoreData, 104
instance, 104
SeedDB, 103
ch.bfh.gr33nopo55um::enocean::persistence::Store

All, 11
An, 12
Application, 12

- createQuery, 113
- entityManager, 115
- getEntityManager, 114
- getInstance, 114
- instance, 115
- jpaStoreId, 115
- ch::bfh::gr33nopo55um::enocean::persistence::TelegramDBFactory
 - logger, 117
 - telegramRouter, 116
- ch::bfh::gr33nopo55um::enocean::telegram::CommonCommand
 - ccCode, 20
 - ccData, 20
 - decodeTelegramData, 17
 - decodeTelegramHeader, 18
 - dumpData, 18
 - encodeTelegram, 18
 - encodeTelegramData, 19
 - getCcCode, 19
 - getCcData, 19
 - id, 20
 - setCcCode, 19
 - setCcData, 19
 - toString, 20
- ch::bfh::gr33nopo55um::enocean::telegram::Dummy
 - decodeTelegramData, 26
 - dumpData, 26
 - encodeTelegram, 26
 - encodeTelegramData, 26
 - id, 27
 - toString, 27
- ch::bfh::gr33nopo55um::enocean::telegram::EncodeDecode
 - decodeTelegram, 29
 - decodeTelegramData, 29
 - decodeTelegramHeader, 30
 - dump, 30
 - dumpData, 30
 - dumpHeader, 31
 - encodeTelegram, 31
 - encodeTelegramData, 31
 - encodeTelegramHeader, 31
 - persist, 32
- ch::bfh::gr33nopo55um::enocean::telegram::Event
 - decodeTelegramData, 37
 - dumpData, 38
 - encodeTelegram, 38
 - encodeTelegramData, 38
 - eventCode, 40
 - EventData, 40
 - getEventCode, 39
 - getEventData, 39
 - getId, 39
 - id, 40
 - setEventCode, 39
 - setEventData, 39
 - setId, 40
- toString, 40
- ch::bfh::gr33nopo55um::enocean::telegram::Radio802dbm, 51
 - decodeTelegramData, 46
 - destinationId, 51
 - dumpData, 47
 - encodeTelegram, 47
 - encodeTelegramData, 48
 - getDbm, 48
 - getDestinationId, 48
 - getId, 48
 - getMessageData, 48
 - getMessageRorg, 49
 - getSourceId, 49
 - id, 51
 - messageData, 52
 - messageRorg, 52
 - setDbm, 49
 - setDestinationId, 49
 - setId, 49
 - setMessageData, 50
 - setMessageRorg, 50
 - setSourceId, 50
 - sourceId, 52
 - toString, 51
- ch::bfh::gr33nopo55um::enocean::telegram::RadioERP1
 - dbm, 60
 - decodeTelegramData, 55
 - destinationID, 60
 - dumpData, 56
 - encodeTelegram, 56
 - encodeTelegramData, 57
 - getDbm, 57
 - getDestinationID, 57
 - getId, 57
 - getSecurityLevel, 57
 - getSubTelNum, 58
 - id, 60
 - securityLevel, 60
 - setDbm, 58
 - setDestinationID, 58
 - setId, 58
 - setSecurityLevel, 59
 - setSubTelNum, 59
 - subTelNum, 60
 - toString, 59
- ch::bfh::gr33nopo55um::enocean::telegram::RadioERP2
 - dbm, 68
 - decodeTelegramData, 64
 - dumpData, 65
 - encodeTelegram, 65
 - encodeTelegramData, 65
 - getDbm, 66
 - getId, 66
 - getRawData, 66
 - getSubTelNum, 66

- id, 68
- rawData, 68
- setDbm, 66
- setId, 67
- setRawData, 67
- setSubTelNum, 67
- subTelNum, 69
- toString, 68
- ch::bfh::gr33nopo55um::enocean::telegram::Radio↔
 - Message
 - dbm, 76
 - decodeTelegramData, 72
 - destinationID, 76
 - dumpData, 73
 - encodeTelegram, 73
 - encodeTelegramData, 73
 - getDbm, 74
 - getDestinationID, 74
 - getId, 74
 - getSourceID, 74
 - id, 76
 - setDbm, 74
 - setDestinationID, 74
 - setId, 75
 - setSourceID, 75
 - sourceID, 76
 - toString, 75
- ch::bfh::gr33nopo55um::enocean::telegram::Radio↔
 - SubTel
 - dbm, 85
 - dbmSubTel, 85
 - decodeTelegramData, 79
 - destinationID, 85
 - dumpData, 80
 - encodeTelegram, 81
 - encodeTelegramData, 81
 - getDbm, 81
 - getDbmSubTel, 81
 - getDestinationID, 81
 - getSecurityLevel, 82
 - getStatusSubTel, 82
 - getSubTelNum, 82
 - getTickSubTel, 82
 - getTimeStamp, 82
 - id, 86
 - securityLevel, 86
 - setDbm, 82
 - setDbmSubTel, 82
 - setDestinationID, 83
 - setSecurityLevel, 83
 - setStatusSubTel, 83
 - setSubTelNum, 84
 - setTickSubTel, 84
 - setTimeStamp, 84
 - statusSubTel, 86
 - subTelNum, 86
 - tickSubTel, 86
 - timeStamp, 86
- toString, 85
- ch::bfh::gr33nopo55um::enocean::telegram::Remote↔
 - ManCommand
 - dbm, 95
 - decodeTelegramData, 91
 - destinationID, 95
 - dumpData, 92
 - encodeTelegram, 92
 - encodeTelegramData, 92
 - getDbm, 93
 - getDestinationID, 93
 - getSendDelay, 93
 - getSourceID, 93
 - id, 95
 - sendDelay, 95
 - setDbm, 93
 - setDestinationID, 93
 - setSendDelay, 94
 - setSourceID, 94
 - sourceID, 95
 - toString, 94
- ch::bfh::gr33nopo55um::enocean::telegram::Response
 - decodeTelegramData, 99
 - dumpData, 100
 - encodeTelegram, 100
 - encodeTelegramData, 100
 - getReturnCode, 101
 - id, 102
 - returnCode, 102
 - setReturnCode, 101
 - toString, 101
- ch::bfh::gr33nopo55um::enocean::telegram::Smart↔
 - AckCommand
 - decodeTelegramData, 109
 - dumpData, 110
 - encodeTelegram, 110
 - encodeTelegramData, 110
 - getSmartAckCommandData, 111
 - getSmartAckCommandNr, 111
 - id, 112
 - setSmartAckCommandData, 111
 - setSmartAckCommandNr, 111
 - smartAckCommandData, 112
 - smartAckCommandNr, 112
 - toString, 111
- ch::bfh::gr33nopo55um::enocean::telegram::Telegram↔
 - Header
 - crcData, 130
 - crcHeader, 130
 - creationDate, 130
 - data, 130
 - dataLength, 130
 - decodeTelegram, 119
 - decodeTelegramHeader, 120
 - dump, 121
 - dumpHeader, 122
 - encodeTelegram, 122
 - encodeTelegramHeader, 123

- fullTelegram, 130
- getCrcData, 123
- getCrcHeader, 123
- getCreationDate, 124
- getData, 124
- getDataLength, 124
- getFullTelegram, 124
- getOptionalLenght, 124
- getSyncByte, 125
- optionalLenght, 130
- persist, 125
- setCrcData, 125
- setCrcHeader, 126
- setCreationDate, 127
- setData, 127
- setDataLength, 128
- setFullTelegram, 128
- setOptionalLenght, 129
- setSyncByte, 129
- syncByte, 130
- ch::bfh::gr33nopo55um::enoccean::test::Sender
 - main, 105
- ch::bfh::gr33nopo55um::enoccean::test::Tester
 - main, 132
- Checksum, 13
- Class, 14
- CommonCommand, 15
- crc8Table
 - ch::bfh::gr33nopo55um::enoccean::helper::CRC, 22
- crcData
 - ch::bfh::gr33nopo55um::enoccean::telegram::TelegramHeader, 130
- crcHeader
 - ch::bfh::gr33nopo55um::enoccean::telegram::TelegramHeader, 130
- createQuery
 - ch::bfh::gr33nopo55um::enoccean::persistence::Store, 113
- creationDate
 - ch::bfh::gr33nopo55um::enoccean::telegram::TelegramHeader, 130
- data
 - ch::bfh::gr33nopo55um::enoccean::telegram::TelegramHeader, 130
- dataLength
 - ch::bfh::gr33nopo55um::enoccean::telegram::TelegramHeader, 130
- DB, 22
- dbm
 - ch::bfh::gr33nopo55um::enoccean::telegram::Radio802, 51
 - ch::bfh::gr33nopo55um::enoccean::telegram::RadioERP1, 60
 - ch::bfh::gr33nopo55um::enoccean::telegram::RadioERP2, 68
 - ch::bfh::gr33nopo55um::enoccean::telegram::RadioMessage, 76
 - ch::bfh::gr33nopo55um::enoccean::telegram::RadioSubTel, 85
 - ch::bfh::gr33nopo55um::enoccean::telegram::RemoteManCommand, 95
- ch::bfh::gr33nopo55um::enoccean::telegram::RadioSubTel, 85
- ch::bfh::gr33nopo55um::enoccean::telegram::RemoteManCommand, 95
- dbmSubTel
 - ch::bfh::gr33nopo55um::enoccean::telegram::RadioSubTel, 85
- decodeTelegram
 - ch::bfh::gr33nopo55um::enoccean::telegram::EncodeDecode, 29
 - ch::bfh::gr33nopo55um::enoccean::telegram::TelegramHeader, 119
- decodeTelegramData
 - ch::bfh::gr33nopo55um::enoccean::telegram::CommonCommand, 17
 - ch::bfh::gr33nopo55um::enoccean::telegram::Dummy, 26
 - ch::bfh::gr33nopo55um::enoccean::telegram::EncodeDecode, 29
 - ch::bfh::gr33nopo55um::enoccean::telegram::Event, 37
 - ch::bfh::gr33nopo55um::enoccean::telegram::Radio802, 46
 - ch::bfh::gr33nopo55um::enoccean::telegram::RadioERP1, 55
 - ch::bfh::gr33nopo55um::enoccean::telegram::RadioERP2, 64
 - ch::bfh::gr33nopo55um::enoccean::telegram::RadioMessage, 72
 - ch::bfh::gr33nopo55um::enoccean::telegram::RadioSubTel, 79
 - ch::bfh::gr33nopo55um::enoccean::telegram::RemoteManCommand, 91
 - ch::bfh::gr33nopo55um::enoccean::telegram::Response, 99
 - ch::bfh::gr33nopo55um::enoccean::telegram::SmartAckCommand, 109
- decodeTelegramHeader
 - ch::bfh::gr33nopo55um::enoccean::telegram::CommonCommand, 18
 - ch::bfh::gr33nopo55um::enoccean::telegram::EncodeDecode, 30
 - ch::bfh::gr33nopo55um::enoccean::telegram::TelegramHeader, 120
- destinationID
 - ch::bfh::gr33nopo55um::enoccean::telegram::RadioERP1, 60
 - ch::bfh::gr33nopo55um::enoccean::telegram::RadioMessage, 76
 - ch::bfh::gr33nopo55um::enoccean::telegram::RadioSubTel, 85
 - ch::bfh::gr33nopo55um::enoccean::telegram::RemoteManCommand, 95
- destinationId
 - ch::bfh::gr33nopo55um::enoccean::telegram::Radio802, 51
- diagnosis, 23
- Dummy, 24

- dump
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
EncodeDecode, 30
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
TelegramHeader, 121
- dumpData
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
CommonCommand, 18
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
Dummy, 26
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
EncodeDecode, 30
 - ch::bfh::gr33nopo55um::enocean::telegram::Event,
38
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
Radio802, 47
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioERP1, 56
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioERP2, 65
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioMessage, 73
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioSubTel, 80
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RemoteManCommand, 92
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
Response, 100
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
SmartAckCommand, 110
- dumpHeader
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
EncodeDecode, 31
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
TelegramHeader, 122
- ERP1, 33
- ERP2, 34
- EncodeDecode, 28
- encodeTelegram
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
CommonCommand, 18
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
Dummy, 26
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
EncodeDecode, 31
 - ch::bfh::gr33nopo55um::enocean::telegram::Event,
38
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
Radio802, 47
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioERP1, 56
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioERP2, 65
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioMessage, 73
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioSubTel, 81
- ch::bfh::gr33nopo55um::enocean::telegram::↔
RemoteManCommand, 92
- ch::bfh::gr33nopo55um::enocean::telegram::↔
Response, 100
- ch::bfh::gr33nopo55um::enocean::telegram::↔
SmartAckCommand, 110
- encodeTelegramData
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
CommonCommand, 19
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
Dummy, 26
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
EncodeDecode, 31
 - ch::bfh::gr33nopo55um::enocean::telegram::Event,
38
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
Radio802, 48
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioERP1, 57
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioERP2, 65
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioMessage, 73
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioSubTel, 81
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RemoteManCommand, 92
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
Response, 100
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
SmartAckCommand, 110
- encodeTelegramHeader
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
EncodeDecode, 31
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
TelegramHeader, 123
- endLogTime
 - ch::bfh::gr33nopo55um::enocean::helper::Time↔
Logger, 134
- EnocealListener, 32
- entityManager
 - ch::bfh::gr33nopo55um::enocean::persistence::↔
Store, 115
- Event, 35
- eventCode
 - ch::bfh::gr33nopo55um::enocean::telegram::Event,
40
- EventData
 - ch::bfh::gr33nopo55um::enocean::telegram::Event,
40
- fullTelegram
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
TelegramHeader, 130
- getCcCode

- ch::bfh::gr33nopo55um::enoccean::telegram::←
CommonCommand, 19
- getCcData
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
CommonCommand, 19
- getCrcData
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
TelegramHeader, 123
- getCrcHeader
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
TelegramHeader, 123
- getCreationDate
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
TelegramHeader, 124
- getData
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
TelegramHeader, 124
- getDataLength
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
TelegramHeader, 124
- getDbm
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
Radio802, 48
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioERP1, 57
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioERP2, 66
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioMessage, 74
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioSubTel, 81
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RemoteManCommand, 93
- getDbmSubTel
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioSubTel, 81
- getDestinationID
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioERP1, 57
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioMessage, 74
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioSubTel, 81
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RemoteManCommand, 93
- getDestinationId
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
Radio802, 48
- getEntityManager
 - ch::bfh::gr33nopo55um::enoccean::persistence::←
Store, 114
- getEventCode
 - ch::bfh::gr33nopo55um::enoccean::telegram::Event,
39
- getEventData
 - ch::bfh::gr33nopo55um::enoccean::telegram::Event,
39
- getFullTelegram
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
TelegramHeader, 124
- getId
 - ch::bfh::gr33nopo55um::enoccean::telegram::Event,
39
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
Radio802, 48
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioERP1, 57
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioERP2, 66
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioMessage, 74
- getInstance
 - ch::bfh::gr33nopo55um::enoccean::persistence::←
SeedDB, 103
 - ch::bfh::gr33nopo55um::enoccean::persistence::←
Store, 114
- getMessageData
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
Radio802, 48
- getMessageRorg
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
Radio802, 49
- getOperationName
 - ch::bfh::gr33nopo55um::enoccean::helper::Time←
Logger, 135
- getOptionalLenght
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
TelegramHeader, 124
- getRawData
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioERP2, 66
- getReturnCode
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
Response, 101
- getSecurityLevel
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioERP1, 57
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioSubTel, 82
- getSendDelay
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RemoteManCommand, 93
- getSmartAckCommandData
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
SmartAckCommand, 111
- getSmartAckCommandNr
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
SmartAckCommand, 111
- getSourceID
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RadioMessage, 74
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
RemoteManCommand, 93
- getSourceId
 - ch::bfh::gr33nopo55um::enoccean::telegram::←
Radio802, 49

- getStartTime
 - ch::bfh::gr33nopo55um::enocean::helper::Time↔
Logger, 135
- getStatusSubTel
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioSubTel, 82
- getStopTime
 - ch::bfh::gr33nopo55um::enocean::helper::Time↔
Logger, 135
- getSubTelNum
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioERP1, 58
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioERP2, 66
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioSubTel, 82
- getSyncByte
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
TelegramHeader, 125
- getTickSubTel
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioSubTel, 82
- getTimeStamp
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioSubTel, 82
- id
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
CommonCommand, 20
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
Dummy, 27
 - ch::bfh::gr33nopo55um::enocean::telegram::Event,
40
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
Radio802, 51
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioERP1, 60
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioERP2, 68
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioMessage, 76
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioSubTel, 86
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RemoteManCommand, 95
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
Response, 102
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
SmartAckCommand, 112
- inputStream
 - ch::bfh::gr33nopo55um::enocean::helper::Read↔
Config, 88
- insertCoreData
 - ch::bfh::gr33nopo55um::enocean::persistence::↔
SeedDB, 104
- instance
 - ch::bfh::gr33nopo55um::enocean::persistence::↔
SeedDB, 104
- ch::bfh::gr33nopo55um::enocean::persistence::↔
Store, 115
- jpaStoreId
 - ch::bfh::gr33nopo55um::enocean::persistence::↔
Store, 115
- logger
 - ch::bfh::gr33nopo55um::enocean::application::↔
EnocealListener, 33
 - ch::bfh::gr33nopo55um::enocean::helper::Time↔
Logger, 137
 - ch::bfh::gr33nopo55um::enocean::persistence::↔
TelegramDBFactory, 117
- Logic, 41
- Main, 41
- main
 - ch::bfh::gr33nopo55um::enocean::application::↔
Application, 13
 - ch::bfh::gr33nopo55um::enocean::test::Sender,
105
 - ch::bfh::gr33nopo55um::enocean::test::Tester, 132
- messageData
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
Radio802, 52
- messageRorg
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
Radio802, 52
- operationName
 - ch::bfh::gr33nopo55um::enocean::helper::Time↔
Logger, 137
- optionalLenght
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
TelegramHeader, 130
- persist
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
EncodeDecode, 32
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
TelegramHeader, 125
- Physical, 42
- Radio, 42
- Radio802, 44
- RadioERP1, 53
- RadioERP2, 62
- RadioMessage, 70
- RadioSubTel, 77
- rawData
 - ch::bfh::gr33nopo55um::enocean::telegram::↔
RadioERP2, 68
- ReadConfig, 87
- readPropertyValue
 - ch::bfh::gr33nopo55um::enocean::helper::Read↔
Config, 87
- remote, 88
- RemoteManCommand, 89
- Response, 97

- returnCode
 - ch::bfh::gr33nopo55um::enocean::telegram::←
Response, 102
- SMART_ACK_COMMAND, 106
- securityLevel
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioERP1, 60
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioSubTel, 86
- SeedDB, 102
 - ch::bfh::gr33nopo55um::enocean::persistence::←
SeedDB, 103
- sendDelay
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RemoteManCommand, 95
- Sender, 105
- Serveral, 106
- setCcCode
 - ch::bfh::gr33nopo55um::enocean::telegram::←
CommonCommand, 19
- setCcData
 - ch::bfh::gr33nopo55um::enocean::telegram::←
CommonCommand, 19
- setCrcData
 - ch::bfh::gr33nopo55um::enocean::telegram::←
TelegramHeader, 125
- setCrcHeader
 - ch::bfh::gr33nopo55um::enocean::telegram::←
TelegramHeader, 126
- setCreationDate
 - ch::bfh::gr33nopo55um::enocean::telegram::←
TelegramHeader, 127
- setData
 - ch::bfh::gr33nopo55um::enocean::telegram::←
TelegramHeader, 127
- setDataLength
 - ch::bfh::gr33nopo55um::enocean::telegram::←
TelegramHeader, 128
- setDbm
 - ch::bfh::gr33nopo55um::enocean::telegram::←
Radio802, 49
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioERP1, 58
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioERP2, 66
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioMessage, 74
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioSubTel, 82
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RemoteManCommand, 93
- setDbmSubTel
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioSubTel, 82
- setDestinationID
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioERP1, 58
- ch::bfh::gr33nopo55um::enocean::telegram::←
RadioMessage, 74
- ch::bfh::gr33nopo55um::enocean::telegram::←
RadioSubTel, 83
- ch::bfh::gr33nopo55um::enocean::telegram::←
RemoteManCommand, 93
- setDestinationId
 - ch::bfh::gr33nopo55um::enocean::telegram::←
Radio802, 49
- setEventCode
 - ch::bfh::gr33nopo55um::enocean::telegram::Event,
39
- setEventData
 - ch::bfh::gr33nopo55um::enocean::telegram::Event,
39
- setFullTelegram
 - ch::bfh::gr33nopo55um::enocean::telegram::←
TelegramHeader, 128
- setId
 - ch::bfh::gr33nopo55um::enocean::telegram::Event,
40
 - ch::bfh::gr33nopo55um::enocean::telegram::←
Radio802, 49
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioERP1, 58
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioERP2, 67
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioMessage, 75
- setMessageData
 - ch::bfh::gr33nopo55um::enocean::telegram::←
Radio802, 50
- setMessageRorg
 - ch::bfh::gr33nopo55um::enocean::telegram::←
Radio802, 50
- setOperationName
 - ch::bfh::gr33nopo55um::enocean::helper::Time←
Logger, 136
- setOptionalLenght
 - ch::bfh::gr33nopo55um::enocean::telegram::←
TelegramHeader, 129
- setRawData
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioERP2, 67
- setReturnCode
 - ch::bfh::gr33nopo55um::enocean::telegram::←
Response, 101
- setSecurityLevel
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioERP1, 59
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RadioSubTel, 83
- setSendDelay
 - ch::bfh::gr33nopo55um::enocean::telegram::←
RemoteManCommand, 94
- setSmartAckCommandData
 - ch::bfh::gr33nopo55um::enocean::telegram::←
SmartAckCommand, 111

- setSmartAckCommandNr
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
SmartAckCommand, 111
- setSourceID
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioMessage, 75
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RemoteManCommand, 94
- setSourceId
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
Radio802, 50
- setStartTime
 - ch::bfh::gr33nopo55um::enoccean::helper::Time↔
Logger, 136
- setStatusSubTel
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioSubTel, 83
- setStopTime
 - ch::bfh::gr33nopo55um::enoccean::helper::Time↔
Logger, 136
- setSubTelNum
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioERP1, 59
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioERP2, 67
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioSubTel, 84
- setSyncByte
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
TelegramHeader, 129
- setTickSubTel
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioSubTel, 84
- setTimeStamp
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioSubTel, 84
- SmartAckCommand, 107
- smartAckCommandData
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
SmartAckCommand, 112
- smartAckCommandNr
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
SmartAckCommand, 112
- sourceID
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioMessage, 76
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RemoteManCommand, 95
- sourceId
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
Radio802, 52
- startLogTime
 - ch::bfh::gr33nopo55um::enoccean::helper::Time↔
Logger, 136
- startTime
 - ch::bfh::gr33nopo55um::enoccean::helper::Time↔
Logger, 137
- statusSubTel
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioSubTel, 86
- stopTime
 - ch::bfh::gr33nopo55um::enoccean::helper::Time↔
Logger, 137
- Store, 113
- subTelNum
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioERP1, 60
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioERP2, 69
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioSubTel, 86
- syncByte
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
TelegramHeader, 130
- TelegramDBFactory, 115
- TelegramHeader, 117
- telegramRouter
 - ch::bfh::gr33nopo55um::enoccean::persistence::↔
TelegramDBFactory, 116
- Tester, 131
- This, 132
- tickSubTel
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioSubTel, 86
- Time, 133
- TimeLogger, 133
- timeStamp
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioSubTel, 86
- toString
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
CommonCommand, 20
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
Dummy, 27
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
Event, 40
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
Radio802, 51
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioERP1, 59
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioERP2, 68
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioMessage, 75
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RadioSubTel, 85
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
RemoteManCommand, 94
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
Response, 101
 - ch::bfh::gr33nopo55um::enoccean::telegram::↔
SmartAckCommand, 111
- Used, 138
- When, 138