

## EventFinder

Generated by Doxygen 1.8.18



<b>1 Deprecated List</b>	<b>1</b>
<b>2 Namespace Index</b>	<b>3</b>
2.1 Packages	3
<b>3 Hierarchical Index</b>	<b>5</b>
3.1 Class Hierarchy	5
<b>4 Class Index</b>	<b>7</b>
4.1 Class List	7
<b>5 File Index</b>	<b>11</b>
5.1 File List	11
<b>6 Namespace Documentation</b>	<b>13</b>
6.1 Package es	13
6.2 Package es.deusto	13
6.3 Package es.deusto.client	13
6.4 Package es.deusto.client.controller	13
6.5 Package es.deusto.client.main	14
6.6 Package es.deusto.client.windows	14
6.7 Package es.deusto.junit	15
6.8 Package es.deusto.junit.client	15
6.9 Package es.deusto.junit.serialization	15
6.10 Package es.deusto.junit.server	15
6.11 Package es.deusto.serialization	15
6.11.1 Detailed Description	16
6.12 Package es.deusto.server	16
6.13 Package es.deusto.server.dao	16
6.14 Package es.deusto.server.data	16
6.15 Package es.deusto.server.server	17
<b>7 Class Documentation</b>	<b>19</b>
7.1 es.deusto.server.server.AppService Class Reference	19
7.1.1 Constructor & Destructor Documentation	20
7.1.1.1 AppService()	20
7.1.2 Member Function Documentation	20
7.1.2.1 attemptNormalLogin()	20
7.1.2.2 attemptNormalSignup()	20
7.1.2.3 attemptNormalUpdate()	21
7.1.2.4 attemptOrganizerLogin()	21
7.1.2.5 attemptOrganizerSignup()	22
7.1.2.6 attemptOrganizerUpdate()	22
7.1.2.7 createEvent()	23

7.1.2.8 createPost()	23
7.1.2.9 deleteEvent()	24
7.1.2.10 deleteOrganizer()	24
7.1.2.11 deleteUser()	25
7.1.2.12 getRecommendedEvents()	25
7.1.2.13 hello()	26
7.1.2.14 recoverPassword()	26
7.1.2.15 updateUser()	26
7.2 es.deusto.client.windows.ChangePass Class Reference	27
7.2.1 Detailed Description	27
7.2.2 Constructor & Destructor Documentation	27
7.2.2.1 ChangePass()	27
7.3 es.deusto.client.windows.CodeVerification Class Reference	28
7.3.1 Constructor & Destructor Documentation	28
7.3.1.1 CodeVerification()	28
7.4 es.deusto.client.controller.Controller Class Reference	28
7.4.1 Detailed Description	29
7.4.2 Constructor & Destructor Documentation	30
7.4.2.1 Controller() [1/2]	30
7.4.2.2 Controller() [2/2]	30
7.4.3 Member Function Documentation	30
7.4.3.1 attemptEventDelete()	30
7.4.3.2 attemptNormalLogin() [1/2]	31
7.4.3.3 attemptNormalLogin() [2/2]	31
7.4.3.4 attemptNormalLoginOrganizer()	32
7.4.3.5 attemptNormalSignup()	33
7.4.3.6 attemptNormalUpdate()	33
7.4.3.7 attemptOrganizerDelete()	34
7.4.3.8 attemptOrganizerSignup()	34
7.4.3.9 attemptOrganizerUpdate()	35
7.4.3.10 attemptPasswordRecovery()	35
7.4.3.11 attemptUserDelete()	35
7.4.3.12 createEvent()	36
7.4.3.13 createPost()	36
7.4.3.14 getLanguageManager()	37
7.4.3.15 getOrganize()	37
7.4.3.16 getRecommendations()	37
7.4.3.17 getUser()	37
7.4.3.18 setOrganizer()	37
7.4.3.19 setUser()	37
7.5 es.deusto.junit.client.ControllerTest Class Reference	38
7.5.1 Detailed Description	38

7.5.2 Member Function Documentation	38
7.5.2.1 configure()	38
7.5.2.2 setUpChild()	39
7.5.2.3 startUp()	39
7.5.2.4 suite()	39
7.5.2.5 tearDownChild()	39
7.5.2.6 testCreateDeleteEvent()	39
7.5.2.7 testNormalLogin()	39
7.5.2.8 testNormalSignupAndDelete()	39
7.5.2.9 testOrganizerLogin()	39
7.5.2.10 testOrganizerSignupAndDelete()	40
7.5.2.11 testUserUpdate()	40
7.5.3 Member Data Documentation	40
7.5.3.1 rule	40
7.6 es.deusto.client.windows.CreateEvent Class Reference	40
7.6.1 Detailed Description	40
7.6.2 Constructor & Destructor Documentation	41
7.6.2.1 CreateEvent()	41
7.6.3 Member Function Documentation	41
7.6.3.1 actionPerformed()	41
7.7 es.deusto.client.windows.CreateOrganizerWindow Class Reference	41
7.7.1 Detailed Description	41
7.7.2 Constructor & Destructor Documentation	42
7.7.2.1 CreateOrganizerWindow()	42
7.7.3 Member Function Documentation	42
7.7.3.1 actionPerformed()	42
7.8 es.deusto.client.windows.CreateUserWindow Class Reference	42
7.8.1 Detailed Description	42
7.8.2 Constructor & Destructor Documentation	43
7.8.2.1 CreateUserWindow()	43
7.8.3 Member Function Documentation	43
7.8.3.1 actionPerformed()	43
7.9 es.deusto.server.dao.DAOFactory Class Reference	43
7.9.1 Detailed Description	44
7.9.2 Member Function Documentation	44
7.9.2.1 closeDAO() [1/5]	44
7.9.2.2 closeDAO() [2/5]	44
7.9.2.3 closeDAO() [3/5]	44
7.9.2.4 closeDAO() [4/5]	44
7.9.2.5 closeDAO() [5/5]	44
7.9.2.6 createEventDAO()	45
7.9.2.7 createOrganizerDAO()	45

7.9.2.8 createPostDAO()	45
7.9.2.9 createTopicDAO()	45
7.9.2.10 createUserDAO()	45
7.9.2.11 getInstance()	45
7.10 es.deusto.server.data.Event Class Reference	45
7.10.1 Detailed Description	46
7.10.2 Constructor & Destructor Documentation	46
7.10.2.1 Event() [1/2]	46
7.10.2.2 Event() [2/2]	46
7.10.3 Member Function Documentation	47
7.10.3.1 addInterested()	47
7.10.3.2 getCity()	47
7.10.3.3 getCountry()	47
7.10.3.4 getDate()	47
7.10.3.5 getDescription()	47
7.10.3.6 getInterested()	47
7.10.3.7 getName()	47
7.10.3.8 getOrganizer()	48
7.10.3.9 getPosts()	48
7.10.3.10 getTopic()	48
7.10.3.11 reduceInterested()	48
7.10.3.12 setCity()	48
7.10.3.13 setCountry()	48
7.10.3.14 setDate()	48
7.10.3.15 setDescription()	49
7.10.3.16 setInterested()	49
7.10.3.17 setName()	49
7.10.3.18 setOrganizer()	49
7.10.3.19 setPosts()	49
7.10.3.20 setTopic()	49
7.10.3.21 toString()	49
7.11 es.deusto.server.dao.EventDAO Class Reference	50
7.11.1 Detailed Description	50
7.11.2 Constructor & Destructor Documentation	50
7.11.2.1 EventDAO()	50
7.11.3 Member Function Documentation	50
7.11.3.1 close()	51
7.11.3.2 deleteEvent()	51
7.11.3.3 getEvents() [1/2]	51
7.11.3.4 getEvents() [2/2]	51
7.11.3.5 getEventsByOrganizer()	51
7.11.3.6 getEventsbyUser()	52

7.11.3.7 getPersistenceManager()	52
7.11.3.8 storeEvent()	52
7.11.3.9 updateEvent()	53
7.12 es.deusto.serialization.EventInfo Class Reference	53
7.12.1 Detailed Description	53
7.12.2 Constructor & Destructor Documentation	54
7.12.2.1 EventInfo() [1/2]	54
7.12.2.2 EventInfo() [2/2]	54
7.12.3 Member Function Documentation	54
7.12.3.1 addInterested()	54
7.12.3.2 getCity()	54
7.12.3.3 getCountry()	54
7.12.3.4 getDate()	54
7.12.3.5 getDescription()	55
7.12.3.6 getInterested()	55
7.12.3.7 getName()	55
7.12.3.8 getOrganizerEmail()	55
7.12.3.9 getPosts()	55
7.12.3.10 getTopic()	55
7.12.3.11 reduceInterested()	55
7.12.3.12 setCity()	55
7.12.3.13 setCountry()	56
7.12.3.14 setDate()	56
7.12.3.15 setDescription()	56
7.12.3.16 setInterested()	56
7.12.3.17 setName()	56
7.12.3.18 setOrganizerEmail()	56
7.12.3.19 setPosts()	56
7.12.3.20 setTopic()	57
7.13 es.deusto.client.windows.EventListItem Class Reference	57
7.13.1 Detailed Description	57
7.13.2 Constructor & Destructor Documentation	57
7.13.2.1 EventListItem()	57
7.14 es.deusto.client.windows.EventOrganizerWindow Class Reference	58
7.14.1 Detailed Description	58
7.14.2 Constructor & Destructor Documentation	58
7.14.2.1 EventOrganizerWindow()	58
7.15 es.deusto.client.windows.EventWindow Class Reference	58
7.15.1 Detailed Description	59
7.15.2 Constructor & Destructor Documentation	59
7.15.2.1 EventWindow()	59
7.16 es.deusto.client.windows.ForgottenPassword Class Reference	59

7.16.1 Detailed Description	59
7.16.2 Constructor & Destructor Documentation	60
7.16.2.1 ForgottenPassword()	60
7.17 es.deusto.server.data.GenericUser Class Reference	60
7.17.1 Detailed Description	60
7.17.2 Member Function Documentation	60
7.17.2.1 getEmail()	61
7.17.2.2 getName()	61
7.17.2.3 getPassword()	61
7.17.2.4 setEmail()	61
7.17.2.5 setName()	61
7.17.2.6 setPassword()	61
7.17.2.7 toString()	61
7.18 es.deusto.serialization.GenericUserInfo Class Reference	62
7.18.1 Detailed Description	62
7.19 es.deusto.client.windows.LanguageManager Class Reference	62
7.19.1 Detailed Description	62
7.19.2 Constructor & Destructor Documentation	62
7.19.2.1 LanguageManager() [1/2]	63
7.19.2.2 LanguageManager() [2/2]	63
7.19.3 Member Function Documentation	63
7.19.3.1 getLanguage()	63
7.19.3.2 getString()	63
7.19.3.3 setLanguage()	64
7.20 es.deusto.serialization.LoginAttempt Class Reference	64
7.20.1 Detailed Description	64
7.20.2 Constructor & Destructor Documentation	64
7.20.2.1 LoginAttempt() [1/2]	65
7.20.2.2 LoginAttempt() [2/2]	65
7.20.3 Member Function Documentation	65
7.20.3.1 getEmail()	65
7.20.3.2 getPassword()	65
7.20.3.3 isOrganizer()	65
7.20.3.4 setEmail()	66
7.20.3.5 setOrganizer()	66
7.20.3.6 setPassword()	66
7.20.3.7 toString()	66
7.21 es.deusto.client.windows.LogInWindow Class Reference	66
7.21.1 Detailed Description	67
7.21.2 Constructor & Destructor Documentation	67
7.21.2.1 LogInWindow()	67
7.21.3 Member Data Documentation	67



7.21.3.1 bCreate	67
7.21.3.2 cBoxAdmin	67
7.21.3.3 id	67
7.22 es.deusto.client.main.Main Class Reference	68
7.22.1 Detailed Description	68
7.22.2 Member Function Documentation	68
7.22.2.1 main()	68
7.23 es.deusto.server.data.Organizer Class Reference	68
7.23.1 Detailed Description	69
7.23.2 Constructor & Destructor Documentation	69
7.23.2.1 Organizer()	69
7.23.3 Member Function Documentation	69
7.23.3.1 addCreatedEvent()	69
7.23.3.2 getCreatedEvents()	69
7.23.3.3 getOrganization()	69
7.23.3.4 setCreatedEvents()	69
7.23.3.5 setOrganization()	70
7.23.3.6 toString()	70
7.24 es.deusto.server.dao.OrganizerDAO Class Reference	70
7.24.1 Detailed Description	70
7.24.2 Constructor & Destructor Documentation	71
7.24.2.1 OrganizerDAO()	71
7.24.3 Member Function Documentation	71
7.24.3.1 close()	71
7.24.3.2 deleteOrganizer()	71
7.24.3.3 getOrganizer()	71
7.24.3.4 getPersistenceManager()	71
7.24.3.5 storeOrganizer()	71
7.24.3.6 updateOrganizer()	72
7.25 es.deusto.client.windows.OrganizerEvents Class Reference	72
7.25.1 Detailed Description	72
7.25.2 Constructor & Destructor Documentation	72
7.25.2.1 OrganizerEvents()	72
7.26 es.deusto.client.windows.OrganizerHome Class Reference	73
7.26.1 Detailed Description	73
7.26.2 Constructor & Destructor Documentation	73
7.26.2.1 OrganizerHome()	73
7.27 es.deusto.serialization.OrganizerInfo Class Reference	73
7.27.1 Detailed Description	74
7.27.2 Constructor & Destructor Documentation	74
7.27.2.1 OrganizerInfo() [1/2]	74
7.27.2.2 OrganizerInfo() [2/2]	74

7.27.3 Member Function Documentation	74
7.27.3.1 getCreatedEvents()	74
7.27.3.2 getEmail()	75
7.27.3.3 getName()	75
7.27.3.4 getOrganization()	75
7.27.3.5 setCreatedEvents()	75
7.27.3.6 setEmail()	75
7.27.3.7 setName()	75
7.27.3.8 setOrganization()	75
7.28 es.deusto.server.data.Post Class Reference	76
7.28.1 Detailed Description	76
7.28.2 Constructor & Destructor Documentation	76
7.28.2.1 Post() [1/2]	76
7.28.2.2 Post() [2/2]	76
7.28.3 Member Function Documentation	76
7.28.3.1 getDate()	77
7.28.3.2 getDescription()	77
7.28.3.3 getEventName()	77
7.28.3.4 getOrganizerEmail()	77
7.28.3.5 getTitle()	77
7.28.3.6 setDate()	77
7.28.3.7 setDescription()	77
7.28.3.8 setEventName()	78
7.28.3.9 setOrganizerEmail()	78
7.28.3.10 setTitle()	78
7.29 es.deusto.client.windows.PostCreationWindow Class Reference	78
7.29.1 Detailed Description	78
7.29.2 Constructor & Destructor Documentation	79
7.29.2.1 PostCreationWindow()	79
7.30 es.deusto.server.dao.PostDAO Class Reference	79
7.30.1 Detailed Description	79
7.30.2 Constructor & Destructor Documentation	79
7.30.2.1 PostDAO()	80
7.30.3 Member Function Documentation	80
7.30.3.1 close()	80
7.30.3.2 deletePost()	80
7.30.3.3 getPersistenceManager()	80
7.30.3.4 getPostsByEvent()	80
7.30.3.5 storePost()	80
7.30.3.6 updatePost()	81
7.31 es.deusto.serialization.PostInfo Class Reference	81
7.31.1 Detailed Description	81

7.31.2 Constructor & Destructor Documentation	81
7.31.2.1 PostInfo() [1/2]	81
7.31.2.2 PostInfo() [2/2]	82
7.31.3 Member Function Documentation	82
7.31.3.1 getDate()	82
7.31.3.2 getDescription()	82
7.31.3.3 getEventName()	82
7.31.3.4 getOrganizerEmail()	82
7.31.3.5 getTitle()	82
7.31.3.6 setDate()	82
7.31.3.7 setDescription()	83
7.31.3.8 setEventName()	83
7.31.3.9 setOrganizerEmail()	83
7.31.3.10 setTitle()	83
7.32 es.deusto.client.windows.Profile Class Reference	83
7.32.1 Detailed Description	84
7.32.2 Constructor & Destructor Documentation	84
7.32.2.1 Profile()	84
7.33 es.deusto.client.windows.SavedEventsWindow Class Reference	84
7.33.1 Detailed Description	84
7.33.2 Constructor & Destructor Documentation	84
7.33.2.1 SavedEventsWindow()	85
7.34 es.deusto.junit.serialization.SerializationTest Class Reference	85
7.34.1 Detailed Description	85
7.34.2 Member Function Documentation	85
7.34.2.1 beforeClass()	86
7.34.2.2 startUp()	86
7.34.2.3 suite()	86
7.34.2.4 testEventInfoFromEvent()	86
7.34.2.5 testEventInfoSetters()	86
7.34.2.6 testOrganizerInfoFromOrganizer()	86
7.34.2.7 testOrganizerInfoSetters()	86
7.34.2.8 testPostInfoFromPost()	86
7.34.2.9 testPostInfoSetters()	87
7.34.2.10 testSignupAttemptOrganizer()	87
7.34.2.11 testSignupAttemptUser()	87
7.34.2.12 testUserInfoFromUser()	87
7.34.2.13 testUserInfoSetters()	87
7.34.2.14 testUserInfoStringConstructor()	87
7.35 es.deusto.server.server.Server Class Reference	87
7.35.1 Detailed Description	88
7.35.2 Constructor & Destructor Documentation	88

7.35.2.1 Server()	88
7.35.3 Member Function Documentation	89
7.35.3.1 attemptNormalLogin()	89
7.35.3.2 attemptNormalSignup()	89
7.35.3.3 attemptNormalUpdate()	90
7.35.3.4 attemptOrganizerDelete()	90
7.35.3.5 attemptOrganizerLogin()	91
7.35.3.6 attemptOrganizerSignup()	91
7.35.3.7 attemptOrganizerUpdate()	92
7.35.3.8 attemptUserDelete()	92
7.35.3.9 createEvent()	93
7.35.3.10 createPost()	93
7.35.3.11 deleteEvent()	94
7.35.3.12 getRecommendedEvents()	94
7.35.3.13 recoverPassword()	95
7.35.3.14 sayHello()	95
7.36 es.deusto.junit.server.ServerTest Class Reference	95
7.36.1 Detailed Description	96
7.36.2 Member Function Documentation	96
7.36.2.1 setUp()	96
7.36.2.2 startUp()	96
7.36.2.3 suite()	96
7.36.2.4 tearDown()	97
7.36.2.5 testCreateEvent()	97
7.36.2.6 testCreatePost()	97
7.36.2.7 testNormalSignup()	97
7.36.2.8 testNormalUpdate()	97
7.36.2.9 testOrganizerLogin()	97
7.36.2.10 testOrganizerSignup()	97
7.36.2.11 testOrganizerUpdate()	98
7.37 es.deusto.serialization.SignupAttempt Class Reference	98
7.37.1 Detailed Description	98
7.37.2 Constructor & Destructor Documentation	98
7.37.2.1 SignupAttempt() [1/2]	99
7.37.2.2 SignupAttempt() [2/2]	99
7.37.3 Member Function Documentation	99
7.37.3.1 buildOrganizer()	99
7.37.3.2 buildUser()	99
7.37.3.3 getCity()	99
7.37.3.4 getCountry()	100
7.37.3.5 getEmail()	100
7.37.3.6 getInterests()	100

7.37.3.7 getName()	100
7.37.3.8 getOrganization()	100
7.37.3.9 getPassword()	100
7.37.3.10 getSavedEvents()	100
7.37.3.11 setCity()	100
7.37.3.12 setCountry()	101
7.37.3.13 setEmail()	101
7.37.3.14 setInterests()	101
7.37.3.15 setName()	101
7.37.3.16 setOrganization()	101
7.37.3.17 setPassword()	101
7.37.3.18 setSaveEvents()	101
7.37.3.19 toString()	102
7.38 es.deusto.server.data.Topic Class Reference	102
7.38.1 Detailed Description	102
7.38.2 Constructor & Destructor Documentation	102
7.38.2.1 Topic() [1/2]	102
7.38.2.2 Topic() [2/2]	102
7.38.3 Member Function Documentation	102
7.38.3.1 getName()	103
7.38.3.2 setName()	103
7.38.3.3 toString()	103
7.39 es.deusto.server.dao.TopicDAO Class Reference	103
7.39.1 Detailed Description	103
7.39.2 Constructor & Destructor Documentation	104
7.39.2.1 TopicDAO()	104
7.39.3 Member Function Documentation	104
7.39.3.1 close()	104
7.39.3.2 deleteTopic()	104
7.39.3.3 getPersistenceManager()	104
7.39.3.4 getTopic()	104
7.39.3.5 storeTopic()	104
7.39.3.6 updateTopic()	105
7.40 es.deusto.serialization.TopicInfo Class Reference	105
7.40.1 Detailed Description	105
7.40.2 Constructor & Destructor Documentation	105
7.40.2.1 TopicInfo() [1/3]	105
7.40.2.2 TopicInfo() [2/3]	106
7.40.2.3 TopicInfo() [3/3]	106
7.40.3 Member Function Documentation	106
7.40.3.1 getName()	106
7.40.3.2 parseTopic()	106

7.40.3.3 setName()	106
7.40.3.4 toString()	107
7.41 es.deusto.server.data.User Class Reference	107
7.41.1 Detailed Description	107
7.41.2 Member Function Documentation	107
7.41.2.1 addEvent()	108
7.41.2.2 addInterest()	108
7.41.2.3 getCity()	108
7.41.2.4 getCountry()	108
7.41.2.5 getInterests()	108
7.41.2.6 getSavedEvents()	108
7.41.2.7 setCity()	108
7.41.2.8 setCountry()	109
7.41.2.9 setInterests()	109
7.41.2.10 setSavedEvents()	109
7.41.2.11 toString()	109
7.42 es.deusto.server.dao.UserDAO Class Reference	109
7.42.1 Detailed Description	110
7.42.2 Constructor & Destructor Documentation	110
7.42.2.1 UserDAO()	110
7.42.3 Member Function Documentation	110
7.42.3.1 close()	110
7.42.3.2 deleteUser()	110
7.42.3.3 getAllUsers()	110
7.42.3.4 getPersistenceManager()	110
7.42.3.5 getUser()	111
7.42.3.6 storeUser()	111
7.42.3.7 updateUser()	111
7.43 es.deusto.client.windows.UserEventsWindow Class Reference	111
7.43.1 Detailed Description	111
7.43.2 Constructor & Destructor Documentation	112
7.43.2.1 UserEventsWindow()	112
7.44 es.deusto.serialization.UserInfo Class Reference	112
7.44.1 Detailed Description	113
7.44.2 Constructor & Destructor Documentation	113
7.44.2.1 UserInfo() [1/3]	113
7.44.2.2 UserInfo() [2/3]	113
7.44.2.3 UserInfo() [3/3]	113
7.44.3 Member Function Documentation	113
7.44.3.1 addInterest()	113
7.44.3.2 getCity()	114
7.44.3.3 getCountry()	114

7.44.3.4 getEmail()	114
7.44.3.5 getInterests()	114
7.44.3.6 getName()	114
7.44.3.7 getSavedEvents()	114
7.44.3.8 setCity()	114
7.44.3.9 setCountry()	115
7.44.3.10 setEmail()	115
7.44.3.11 setInterests()	115
7.44.3.12 setName()	115
7.44.3.13 setSavedEvents()	115
<b>8 File Documentation</b>	<b>117</b>
8.1 src/main/java/es/deusto/client/controller/Controller.java File Reference	117
8.2 src/main/java/es/deusto/client/main/Main.java File Reference	117
8.3 src/main/java/es/deusto/client/windows/ChangePass.java File Reference	117
8.4 src/main/java/es/deusto/client/windows/CodeVerification.java File Reference	118
8.5 src/main/java/es/deusto/client/windows/CreateEvent.java File Reference	118
8.6 src/main/java/es/deusto/client/windows/CreateOrganizerWindow.java File Reference	118
8.7 src/main/java/es/deusto/client/windows/CreateUserWindow.java File Reference	119
8.8 src/main/java/es/deusto/client/windows/EventListItem.java File Reference	119
8.9 src/main/java/es/deusto/client/windows/EventOrganizerWindow.java File Reference	119
8.10 src/main/java/es/deusto/client/windows/EventWindow.java File Reference	119
8.11 src/main/java/es/deusto/client/windows/ForgottenPassword.java File Reference	120
8.12 src/main/java/es/deusto/client/windows/LanguageManager.java File Reference	120
8.13 src/main/java/es/deusto/client/windows/LoginWindow.java File Reference	120
8.14 src/main/java/es/deusto/client/windows/OrganizerEvents.java File Reference	121
8.15 src/main/java/es/deusto/client/windows/OrganizerHome.java File Reference	121
8.16 src/main/java/es/deusto/client/windows/PostCreationWindow.java File Reference	121
8.17 src/main/java/es/deusto/client/windows/Profile.java File Reference	121
8.18 src/main/java/es/deusto/client/windows/SavedEventsWindow.java File Reference	122
8.19 src/main/java/es/deusto/client/windows/UserEventsWindow.java File Reference	122
8.20 src/main/java/es/deusto/serialization/EventInfo.java File Reference	122
8.21 src/main/java/es/deusto/serialization/GenericUserInfo.java File Reference	123
8.22 src/main/java/es/deusto/serialization/LoginAttempt.java File Reference	123
8.23 src/main/java/es/deusto/serialization/OrganizerInfo.java File Reference	123
8.24 src/main/java/es/deusto/serialization/PostInfo.java File Reference	123
8.25 src/main/java/es/deusto/serialization/SignupAttempt.java File Reference	124
8.26 src/main/java/es/deusto/serialization/TopicInfo.java File Reference	124
8.27 src/main/java/es/deusto/serialization/UserInfo.java File Reference	124
8.28 src/main/java/es/deusto/server/dao/DAOFactory.java File Reference	124
8.29 src/main/java/es/deusto/server/dao/EventDAO.java File Reference	125
8.30 src/main/java/es/deusto/server/dao/OrganizerDAO.java File Reference	125

8.31 src/main/java/es/deusto/server/dao/PostDAO.java File Reference . . . . .	125
8.32 src/main/java/es/deusto/server/dao/TopicDAO.java File Reference . . . . .	125
8.33 src/main/java/es/deusto/server/dao/UserDAO.java File Reference . . . . .	126
8.34 src/main/java/es/deusto/server/data/Event.java File Reference . . . . .	126
8.35 src/main/java/es/deusto/server/data/GenericUser.java File Reference . . . . .	126
8.36 src/main/java/es/deusto/server/data/Organizer.java File Reference . . . . .	126
8.37 src/main/java/es/deusto/server/data/Post.java File Reference . . . . .	127
8.38 src/main/java/es/deusto/server/data/Topic.java File Reference . . . . .	127
8.39 src/main/java/es/deusto/server/data/User.java File Reference . . . . .	127
8.40 src/main/java/es/deusto/server/server/AppService.java File Reference . . . . .	127
8.41 src/main/java/es/deusto/server/server/Server.java File Reference . . . . .	128
8.42 src/test/java/es/deusto/junit/client/ControllerTest.java File Reference . . . . .	128
8.43 src/test/java/es/deusto/junit/serialization/SerializationTest.java File Reference . . . . .	128
8.44 src/test/java/es/deusto/junit/server/ServerTest.java File Reference . . . . .	128

<b>Index</b>	<b>129</b>
--------------	------------



## Chapter 1

# Deprecated List

**Member** [es.deusto.client.controller.Controller.attemptNormalLogin](#) (String email, String password, boolean organizer)

beeter to use attemptNormalLogin(email, password) and attemptNormalLoginOrganizer !

**Member** [es.deusto.server.server.AppService.updateUser](#) ()

no useful for normal use.

**Member** [es.deusto.server.server.AppService.hello](#) ()

**Member** [es.deusto.server.server.Server.sayHello](#) ()



## Chapter 2

# Namespace Index

### 2.1 Packages

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

<a href="#">es</a>	13
<a href="#">es.deusto</a>	13
<a href="#">es.deusto.client</a>	13
<a href="#">es.deusto.client.controller</a>	13
<a href="#">es.deusto.client.main</a>	14
<a href="#">es.deusto.client.windows</a>	14
<a href="#">es.deusto.junit</a>	15
<a href="#">es.deusto.junit.client</a>	15
<a href="#">es.deusto.junit.serialization</a>	15
<a href="#">es.deusto.junit.server</a>	15
<a href="#">es.deusto.serialization</a>	15
<a href="#">es.deusto.server</a>	16
<a href="#">es.deusto.server.dao</a>	16
<a href="#">es.deusto.server.data</a>	16
<a href="#">es.deusto.server.server</a>	17



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

es.deusto.server.server.AppService . . . . .	19
es.deusto.client.controller.Controller . . . . .	28
es.deusto.server.dao.DAOFactory . . . . .	43
es.deusto.server.data.Event . . . . .	45
es.deusto.server.dao.EventDAO . . . . .	50
es.deusto.serialization.EventInfo . . . . .	53
es.deusto.server.data.GenericUser . . . . .	60
es.deusto.server.data.Organizer . . . . .	68
es.deusto.server.data.User . . . . .	107
es.deusto.serialization.GenericUserInfo . . . . .	62
es.deusto.serialization.OrganizerInfo . . . . .	73
es.deusto.serialization.UserInfo . . . . .	112
es.deusto.client.windows.LanguageManager . . . . .	62
es.deusto.serialization.LoginAttempt . . . . .	64
es.deusto.client.main.Main . . . . .	68
es.deusto.server.dao.OrganizerDAO . . . . .	70
es.deusto.server.data.Post . . . . .	76
es.deusto.server.dao.PostDAO . . . . .	79
es.deusto.serialization.PostInfo . . . . .	81
es.deusto.junit.serialization.SerializationTest . . . . .	85
es.deusto.server.server.Server . . . . .	87
es.deusto.junit.server.ServerTest . . . . .	95
es.deusto.serialization.SignupAttempt . . . . .	98
es.deusto.server.data.Topic . . . . .	102
es.deusto.server.dao.TopicDAO . . . . .	103
es.deusto.serialization.TopicInfo . . . . .	105
es.deusto.server.dao.UserDAO . . . . .	109
ActionListener	
es.deusto.client.windows.CreateEvent . . . . .	40
es.deusto.client.windows.CreateOrganizerWindow . . . . .	41
es.deusto.client.windows.CreateUserWindow . . . . .	42
JerseyTest	
es.deusto.junit.client.ControllerTest . . . . .	38
JFrame	
es.deusto.client.windows.ChangePass . . . . .	27

es.deusto.client.windows.CodeVerification . . . . .	28
es.deusto.client.windows.CreateEvent . . . . .	40
es.deusto.client.windows.CreateOrganizerWindow . . . . .	41
es.deusto.client.windows.CreateUserWindow . . . . .	42
es.deusto.client.windows.EventOrganizerWindow . . . . .	58
es.deusto.client.windows.EventWindow . . . . .	58
es.deusto.client.windows.ForgottenPassword . . . . .	59
es.deusto.client.windows.LogInWindow . . . . .	66
es.deusto.client.windows.OrganizerEvents . . . . .	72
es.deusto.client.windows.OrganizerHome . . . . .	73
es.deusto.client.windows.PostCreationWindow . . . . .	78
es.deusto.client.windows.Profile . . . . .	83
es.deusto.client.windows.SavedEventsWindow . . . . .	84
es.deusto.client.windows.UserEventsWindow . . . . .	111
JPanel	
es.deusto.client.windows.EventListItem . . . . .	57

## Chapter 4

# Class Index

### 4.1 Class List

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

<a href="#">es.deusto.server.server.AppService</a>	19
<a href="#">es.deusto.client.windows.ChangePass</a>	
This class is the one used to create a window to change the user password	27
<a href="#">es.deusto.client.windows.CodeVerification</a>	28
<a href="#">es.deusto.client.controller.Controller</a>	
This class is the one carrying out all the communication with the server	28
<a href="#">es.deusto.junit.client.ControllerTest</a>	
This classes is the one that tests the controller component to see if it is working as intended	38
<a href="#">es.deusto.client.windows.CreateEvent</a>	
This class is the one used to create a window that it allows the user to create a new event	40
<a href="#">es.deusto.client.windows.CreateOrganizerWindow</a>	
This class is the one used to create a window that allows to create a new organizer	41
<a href="#">es.deusto.client.windows.CreateUserWindow</a>	
This class is the one used to create a window that allows to create a new user	42
<a href="#">es.deusto.server.dao.DAOFactory</a>	
This class is used to create a DAO factory that is used by all of the other DAO classes, to create them or to close them	43
<a href="#">es.deusto.server.data.Event</a>	
This classes is the one that defines an event and it is used to create one	45
<a href="#">es.deusto.server.dao.EventDAO</a>	
This classes is the one used to create an event DAO	50
<a href="#">es.deusto.serialization.EventInfo</a>	
DTO equivalent class of the Event class from the server	53
<a href="#">es.deusto.client.windows.EventListItem</a>	
This class is the one used to create a window that allows the user to have an event list	57
<a href="#">es.deusto.client.windows.EventOrganizerWindow</a>	
This class is the one used to create a window that allows to an organizer to change or administrate one of his events	58
<a href="#">es.deusto.client.windows.EventWindow</a>	
This class is the one used to create a window that allows to display all the information of one event	58
<a href="#">es.deusto.client.windows.ForgottenPassword</a>	
This class is the one used to create a window that allows a user to log in when he forgets the password	59
<a href="#">es.deusto.server.data.GenericUser</a>	
This classes is the father class for normal user and organizer	60

<a href="#">es.deusto.serialization.GenericUserInfo</a>	
This class works as a Tag for the Classes related to the User and Organizer . . . . .	62
<a href="#">es.deusto.client.windows.LanguageManager</a>	
This class is used to manage the different languages supported by the application . . . . .	62
<a href="#">es.deusto.serialization.LoginAttempt</a>	
This class encapsulates all the information required when a user (Either normal User or Organizer) logs in . . . . .	64
<a href="#">es.deusto.client.windows.LoginWindow</a>	
This class is the one used to create a window that allows a user/organizer to log in to the application . . . . .	66
<a href="#">es.deusto.client.main.Main</a>	
This class is the one that starts the application . . . . .	68
<a href="#">es.deusto.server.data.Organizer</a>	
This classes is the one that defines an organizer and it is used to create one . . . . .	68
<a href="#">es.deusto.server.dao.OrganizerDAO</a>	
This classes is the one used to create an organizer DAO . . . . .	70
<a href="#">es.deusto.client.windows.OrganizerEvents</a>	
This class is the one used to create a window that allows an organizer to have all of his events . . . . .	72
<a href="#">es.deusto.client.windows.OrganizerHome</a>	
This class is the one used to create a window that allows an organizer to have a home window where he can access to all of the functionalities of the application . . . . .	73
<a href="#">es.deusto.serialization.OrganizerInfo</a>	
DTO equivalent class of the Organizer class from the server . . . . .	73
<a href="#">es.deusto.server.data.Post</a>	
This classes is the one that defines a post and it is used to create one . . . . .	76
<a href="#">es.deusto.client.windows.PostCreationWindow</a>	
This class is the one used to create a window that allows an organizer to have like a double check of the event he has just created . . . . .	78
<a href="#">es.deusto.server.dao.PostDAO</a>	
This classes is the one used to create a DAO for a post . . . . .	79
<a href="#">es.deusto.serialization.PostInfo</a>	
DTO equivalent class of the Post class from the server . . . . .	81
<a href="#">es.deusto.client.windows.Profile</a>	
This class is the one used to create a window that allows a user/organizer to have a profile with all of his information . . . . .	83
<a href="#">es.deusto.client.windows.SavedEventsWindow</a>	
This class is the one used to create a window that allows to scroll, search and choose from all of the events stored in the database . . . . .	84
<a href="#">es.deusto.junit.serialization.SerializationTest</a>	
This classes is the one that tests funcionalities such as creating an event, creating a user, orginzer, crea ting posts and topics . . . . .	85
<a href="#">es.deusto.server.server.Server</a>	
This class is the one that receives the requests from the client . . . . .	87
<a href="#">es.deusto.junit.server.ServerTest</a>	
This classes is used to create several tests to see if the server is working as intended . . . . .	95
<a href="#">es.deusto.serialization.SignupAttempt</a>	
This class encapsulates all the information required when a user (Either normal User or Organizer) signs up . . . . .	98
<a href="#">es.deusto.server.data.Topic</a>	
This classes is the one that defines a topic and it is used to create one . . . . .	102
<a href="#">es.deusto.server.dao.TopicDAO</a>	
This classes is the one used to create a DAO a topic . . . . .	103
<a href="#">es.deusto.serialization.TopicInfo</a>	
DTO equivalent class of the Topic class from the server . . . . .	105
<a href="#">es.deusto.server.data.User</a>	
This classes is the one that defines a normal user and it is used to create one . . . . .	107
<a href="#">es.deusto.server.dao.UserDAO</a>	
This classes is the one used to create an user DAO . . . . .	109



[es.deusto.client.windows.UserEventsWindow](#)

This class is the one used to create a window that allows a user to scroll, search and select from a list of events he previously has chosen from . . . . . 111

[es.deusto.serialization.UserInfo](#)

DTO equivalent class of the User class from the server . . . . . 112



## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

src/main/java/es/deusto/client/controller/Controller.java	117
src/main/java/es/deusto/client/main/Main.java	117
src/main/java/es/deusto/client/windows/ChangePass.java	117
src/main/java/es/deusto/client/windows/CodeVerification.java	118
src/main/java/es/deusto/client/windows/CreateEvent.java	118
src/main/java/es/deusto/client/windows/CreateOrganizerWindow.java	118
src/main/java/es/deusto/client/windows/CreateUserWindow.java	119
src/main/java/es/deusto/client/windows/EventListItem.java	119
src/main/java/es/deusto/client/windows/EventOrganizerWindow.java	119
src/main/java/es/deusto/client/windows/EventWindow.java	119
src/main/java/es/deusto/client/windows/ForgottenPassword.java	120
src/main/java/es/deusto/client/windows/LanguageManager.java	120
src/main/java/es/deusto/client/windows/LoginWindow.java	120
src/main/java/es/deusto/client/windows/OrganizerEvents.java	121
src/main/java/es/deusto/client/windows/OrganizerHome.java	121
src/main/java/es/deusto/client/windows/PostCreationWindow.java	121
src/main/java/es/deusto/client/windows/Profile.java	121
src/main/java/es/deusto/client/windows/SavedEventsWindow.java	122
src/main/java/es/deusto/client/windows/UserEventsWindow.java	122
src/main/java/es/deusto/serialization/EventInfo.java	122
src/main/java/es/deusto/serialization/GenericUserInfo.java	123
src/main/java/es/deusto/serialization/LoginAttempt.java	123
src/main/java/es/deusto/serialization/OrganizerInfo.java	123
src/main/java/es/deusto/serialization/PostInfo.java	123
src/main/java/es/deusto/serialization/SignupAttempt.java	124
src/main/java/es/deusto/serialization/TopicInfo.java	124
src/main/java/es/deusto/serialization/UserInfo.java	124
src/main/java/es/deusto/server/dao/DAOFactory.java	124
src/main/java/es/deusto/server/dao/EventDAO.java	125
src/main/java/es/deusto/server/dao/OrganizerDAO.java	125
src/main/java/es/deusto/server/dao/PostDAO.java	125
src/main/java/es/deusto/server/dao/TopicDAO.java	125
src/main/java/es/deusto/server/dao/UserDAO.java	126
src/main/java/es/deusto/server/data/Event.java	126
src/main/java/es/deusto/server/data/GenericUser.java	126

src/main/java/es/deusto/server/data/ <a href="#">Organizer.java</a>	126
src/main/java/es/deusto/server/data/ <a href="#">Post.java</a>	127
src/main/java/es/deusto/server/data/ <a href="#">Topic.java</a>	127
src/main/java/es/deusto/server/data/ <a href="#">User.java</a>	127
src/main/java/es/deusto/server/server/ <a href="#">AppService.java</a>	127
src/main/java/es/deusto/server/server/ <a href="#">Server.java</a>	128
src/test/java/es/deusto/junit/client/ <a href="#">ControllerTest.java</a>	128
src/test/java/es/deusto/junit/serialization/ <a href="#">SerializationTest.java</a>	128
src/test/java/es/deusto/junit/server/ <a href="#">ServerTest.java</a>	128

## Chapter 6

# Namespace Documentation

### 6.1 Package es

#### Packages

- package [deusto](#)

### 6.2 Package es.deusto

#### Packages

- package [client](#)
- package [junit](#)
- package [serialization](#)
- package [server](#)

### 6.3 Package es.deusto.client

#### Packages

- package [controller](#)
- package [main](#)
- package [windows](#)

### 6.4 Package es.deusto.client.controller

#### Classes

- class [Controller](#)

*This class is the one carrying out all the communication with the server.*

## 6.5 Package es.deusto.client.main

### Classes

- class [Main](#)

*This class is the one that starts the application.*

## 6.6 Package es.deusto.client.windows

### Classes

- class [ChangePass](#)

*This class is the one used to create a window to change the user password.*

- class [CodeVerification](#)

- class [CreateEvent](#)

*This class is the one used to create a window that it allows the user to create a new event.*

- class [CreateOrganizerWindow](#)

*This class is the one used to create a window that allows to create a new organizer.*

- class [CreateUserWindow](#)

*This class is the one used to create a window that allows to create a new user.*

- class [EventListItem](#)

*This class is the one used to create a window that allows the user to have an event list.*

- class [EventOrganizerWindow](#)

*This class is the one used to create a window that allows to an organizer to change or administrate one of his events.*

- class [EventWindow](#)

*This class is the one used to create a window that allows to display all the information of one event.*

- class [ForgottenPassword](#)

*This class is the one used to create a window that allows a user to log in when he forgets the password.*

- class [LanguageManager](#)

*This class is used to manage the different languages supported by the application.*

- class [LoginWindow](#)

*This class is the one used to create a window that allows a user/organizer to log in to the application.*

- class [OrganizerEvents](#)

*This class is the one used to create a window that allows an organizer to have all of his events.*

- class [OrganizerHome](#)

*This class is the one used to create a window that allows an organizer to have a home window where he can access to all of the functionalities of the application.*

- class [PostCreationWindow](#)

*This class is the one used to create a window that allows an organizer to have like a double check of the event he has just created.*

- class [Profile](#)

*This class is the one used to create a window that allows a user/organizer to have a profile with all of his information.*

- class [SavedEventsWindow](#)

*This class is the one used to create a window that allows to scroll, search and choose from all of the events stored in the database.*

- class [UserEventsWindow](#)

*This class is the one used to create a window that allows a user to scroll, search and select from a list of events he previously has chosen from.*

## 6.7 Package es.deusto.junit

### Packages

- package [client](#)
- package [serialization](#)
- package [server](#)

## 6.8 Package es.deusto.junit.client

### Classes

- class [ControllerTest](#)  
*This classes is the one that tests the controller component to see if it is working as intended.*

## 6.9 Package es.deusto.junit.serialization

### Classes

- class [SerializationTest](#)  
*This classes is the one that tests funcionalities such as creating an event, creating a user, orginzer, crea ting posts and topics.*

## 6.10 Package es.deusto.junit.server

### Classes

- class [ServerTest](#)  
*This classes is used to create several tests to see if the server is working as intended.*

## 6.11 Package es.deusto.serialization

### Classes

- class [EventInfo](#)  
*DTO equivalent class of the Event class from the server.*
- class [GenericUserInfo](#)  
*This class works as a Tag for the Classes related to the User and Organizer.*
- class [LoginAttempt](#)  
*This class encapsulates all the information required when a user (Either normal User or Organizer) logs in.*
- class [OrganizerInfo](#)  
*DTO equivalent class of the Organizer class from the server.*
- class [PostInfo](#)  
*DTO equivalent class of the Post class from the server.*
- class [SignupAttempt](#)  
*This class encapsulates all the information required when a user (Either normal User or Organizer) signs up.*
- class [TopicInfo](#)  
*DTO equivalent class of the Topic class from the server.*
- class [UserInfo](#)  
*DTO equivalent class of the User class from the server.*

### 6.11.1 Detailed Description

All the ...Info classes work as a middleman for the client and server bussines objects. Due to the functionality of this classes their structure is similar to the 'original' ones in the server Examples: [UserInfo](#): User [EventInfo](#): Event Also the ...Info clases are the ones shown in the GUI.

## 6.12 Package es.deusto.server

### Packages

- package [dao](#)
- package [data](#)
- package [server](#)

## 6.13 Package es.deusto.server.dao

### Classes

- class [DAOFactory](#)  
*This class is used to create a DAO factory that is used by all of the other DAO classes, to create them or to close them.*
- class [EventDAO](#)  
*This classes is the one used to create an event DAO.*
- class [OrganizerDAO](#)  
*This classes is the one used to create an organizer DAO.*
- class [PostDAO](#)  
*This classes is the one used to create a DAO for a post.*
- class [TopicDAO](#)  
*This classes is the one used to create a DAO a topic.*
- class [UserDAO](#)  
*This classes is the one used to create an user DAO.*

## 6.14 Package es.deusto.server.data

### Classes

- class [Event](#)  
*This classes is the one that defines an event and it is used to create one.*
- class [GenericUser](#)  
*This classes is the father class for normal user and organizer.*
- class [Organizer](#)  
*This classes is the one that defines an organizer and it is used to create one.*
- class [Post](#)  
*This classes is the one that defines a post and it is used to create one.*
- class [Topic](#)  
*This classes is the one that defines a topic and it is used to create one.*
- class [User](#)  
*This classes is the one that defines a normal user and it is used to create one.*



## 6.15 Package es.deusto.server.server

### Classes

- class [AppService](#)
- class [Server](#)

*This class is the one that receives the requests from the client.*



## Chapter 7

# Class Documentation

### 7.1 es.deusto.server.server.AppService Class Reference

#### Public Member Functions

- [AppService](#) ()  
*This class will include methods representing the different functionalities of the server.*
- [User attemptNormalLogin](#) ([LoginAttempt](#) login)  
*This method is used for the login process of regular users.*
- [Organizer attemptOrganizerLogin](#) ([LoginAttempt](#) login)  
*This method is used for the login process of organizers.*
- [User attemptNormalSignup](#) ([SignupAttempt](#) signup)  
*This method is used for the signup process of regular users.*
- [Organizer attemptOrganizerSignup](#) ([SignupAttempt](#) signup)  
*This method is used for the signup process of organizers.*
- [User attemptNormalUpdate](#) ([SignupAttempt](#) signup)  
*This method is used for the update process of regular users (not organizers).*
- [Organizer attemptOrganizerUpdate](#) ([SignupAttempt](#) signup)  
*This method is used for the updating process of organizers.*
- boolean [deleteUser](#) (String email)  
*This method is used for the delete process of users.*
- boolean [deleteOrganizer](#) (String email)  
*This method is used for the delete process of organizers.*
- [Event createEvent](#) ([EventInfo](#) eventInfo)  
*Method used for storing new Events in the database, used only by the organizers.*
- ArrayList< [EventInfo](#) > [getRecommendedEvents](#) ([SignupAttempt](#) signupAttempt)  
*Creates a list of events that coincide with the interests (Topic) of a user.*
- void [deleteEvent](#) ([EventInfo](#) info)  
*This method is used for event removal.*
- void [recoverPassword](#) ([LoginAttempt](#) login)  
*This method is used for new password generation.*
- [Post createPost](#) ([PostInfo](#) info)  
*This method is used for the creation of new Post from a given Event.*
- void [updateUser](#) ()  
*Used for testing, creates an instance of a user and updates it in the database.*
- void [hello](#) ()  
*Used for testing.*

## 7.1.1 Constructor & Destructor Documentation

### 7.1.1.1 AppService()

```
es.deusto.server.server.AppService.AppService ( )
```

This class will include methods representing the different functionalities of the server.

This class should be called by the [Server](#) class, so that the [Server](#) is abstracted from the business logic of the project. In order to access to the database, use the DAO objects (create them using DAOFactory class).

## 7.1.2 Member Function Documentation

### 7.1.2.1 attemptNormalLogin()

```
User es.deusto.server.server.AppService.attemptNormalLogin (
    LoginAttempt login )
```

This method is used for the login process of regular users.

Receives a LoginAttempt object with the email and password and compares it with the copy stored in the database.

#### Parameters

<i>login</i>	LoginAttempt with the requester's email and password.
--------------	---

#### Returns

User object to which the login email corresponds. If the credentials are invalid, null is returned.

#### Since

Sprint 1

### 7.1.2.2 attemptNormalSignup()

```
User es.deusto.server.server.AppService.attemptNormalSignup (
    SignupAttempt signup )
```

This method is used for the signup process of regular users.

Receives a SignupAttempt object with the information of the user and tries to submit it to the database.

**Parameters**

<i>signup</i>	SignupAttempt with the user's signup data (name, email, password and city).
---------------	---

**Returns**

User object that has been created as a result of the request. If the email is already in use, null is returned.

**Since**

Sprint 1

**7.1.2.3 attemptNormalUpdate()**

```
User es.deusto.server.server.AppService.attemptNormalUpdate (
    SignupAttempt signup )
```

This method is used for the update process of regular users (not organizers).

Receives a SignupAttempt object and tries to update an existing user in the database.

**Parameters**

<i>signup</i>	SignupAttempt with the user's data (name, email, an empty password, city).
---------------	--

**Since**

Sprint 2

**Returns**

User object that has been created as a result of the request. If the email doesn't exist null is returned

**7.1.2.4 attemptOrganizerLogin()**

```
Organizer es.deusto.server.server.AppService.attemptOrganizerLogin (
    LoginAttempt login )
```

This method is used for the login process of organizers.

Receives a LoginAttempt object with the email and password and compares it with the copy stored in the database.

**Parameters**

<i>login</i>	LoginAttempt with the requester's email and password.
--------------	---

**Returns**

Organizer object to which the login email corresponds. If the credentials are invalid, null is returned.

**Since**

Sprint 2

**7.1.2.5 attemptOrganizerSignup()**

```
Organizer es.deusto.server.server.AppService.attemptOrganizerSignup (
    SignupAttempt signup )
```

This method is used for the signup process of organizers.

Receives a SignupAttempt object with the information of the user and tries to submit it to the database.

**Parameters**

<i>signup</i>	SignupAttempt with the organizer's signup data - name, email, password and organization
---------------	---

**Returns**

Organizer object that has been created as a result of the request. If the email is already in use, null is returned.

**Since**

Sprint 2

**7.1.2.6 attemptOrganizerUpdate()**

```
Organizer es.deusto.server.server.AppService.attemptOrganizerUpdate (
    SignupAttempt signup )
```

This method is used for the updating process of organizers.

Receives a SignupAttempt object and tries to update an existing organizer in the database.

**Parameters**

<i>signup</i>	SignupAttempt object with the requester's data (name, email, empty password, organization).
---------------	---

**Since**

Sprint 2

**Returns**

Organizer object that has been updated as a result of the request. If the email doesn't exist null is returned.

**7.1.2.7 createEvent()**

```
Event es.deusto.server.server.AppService.createEvent (
    EventInfo eventInfo )
```

Method used for storing new Events in the database, used only by the organizers.

Receives an EventInfo, transforms it into a Event and stores it in the database.

**Parameters**

<i>eventInfo</i>	
------------------	--

**Returns**

Event object of the newly stored event.

**Since**

sprint 2

**7.1.2.8 createPost()**

```
Post es.deusto.server.server.AppService.createPost (
    PostInfo info )
```

This method is used for the creation of new Post from a given Event.

It is invoked whenever a POST request is made to the following path: /createEvent.

**Parameters**

<i>info</i>	PostInfo object with the data of the event that will be created.
-------------	--

**Returns**

Post object containing the information of the stored result.

**Since**

Sprint 2

**7.1.2.9 deleteEvent()**

```
void es.deusto.server.server.AppService.deleteEvent (
    EventInfo info )
```

This method is used for event removal.

It receives an EventInfo with the data of the event to be deleted.

**Parameters**

<i>info</i>	EventInfo with the data of the event
-------------	--------------------------------------

**Since**

Sprint 3

**7.1.2.10 deleteOrganizer()**

```
boolean es.deusto.server.server.AppService.deleteOrganizer (
    String email )
```

This method is used for the delete process of organizers.

Receives an String with the email and tries to delete the organizer account related to it from the database.

**Parameters**

<i>email</i>	of the organizer account to be deleted.
--------------	---



**Since**

Sprint 2

**Returns**

boolean True if the email belongs to a organizer and the delete is complete successfully, false otherwise.

**7.1.2.11 deleteUser()**

```
boolean es.deusto.server.server.AppService.deleteUser (
    String email )
```

This method is used for the delete process of users.

Receives an String with the email and tries to delete the account related to it from the database.

**Parameters**

<i>email</i>	of the account to be deleted.
--------------	-------------------------------

**Since**

Sprint 2

**Returns**

boolean True if the email belongs to a user and the delete is complete successfully, false otherwise.

**7.1.2.12 getRecommendedEvents()**

```
ArrayList<EventInfo> es.deusto.server.server.AppService.getRecommendedEvents (
    SignupAttempt signupAttempt )
```

Creates a list of events that coincide with the interests (Topic) of a user.

This method reads the interests of a User, and returns the adequate Events from the database. It is used for making automatic recomendations, a user should be able to see a list of events that he/she should be interested.

**Parameters**

<i>signupAttempt</i>	SignupAttempt object containing the interests of the user.
----------------------	--

**Returns**

ArrayList of EventInfo list of events that coincide with the interests of a user.

**Since**

Sprint 3

**7.1.2.13 hello()**

```
void es.deusto.server.server.AppService.hello ( )
```

Used for testing.

**Deprecated****7.1.2.14 recoverPassword()**

```
void es.deusto.server.server.AppService.recoverPassword (
    LoginAttempt login )
```

This method is used for new password generation.

It receives a LoginAttempt with the email of the user that wants to recover his/her password, generates a new password for that user and updates the data in the database. It also sends an email to the received email with the new password so that the user can log in. If there is no account in the database with that same email linked to it, the method does nothing.

**Parameters**

<i>login</i>	LoginAttempt with the user's email
--------------	------------------------------------

**Since**

Sprint 1

**7.1.2.15 updateUser()**

```
void es.deusto.server.server.AppService.updateUser ( )
```

Used for testing, creates an instance of a user and updates it in the database.

**Deprecated** no useful for normal use.

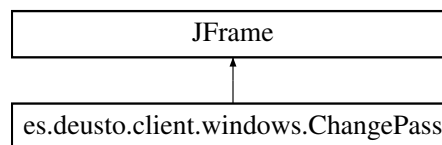
The documentation for this class was generated from the following file:

- [src/main/java/es/deusto/server/server/AppService.java](#)

## 7.2 es.deusto.client.windows.ChangePass Class Reference

This class is the one used to create a window to change the user password.

Inheritance diagram for es.deusto.client.windows.ChangePass:



### Public Member Functions

- [ChangePass](#) ([Controller](#) controller)  
*Create the frame.*

### 7.2.1 Detailed Description

This class is the one used to create a window to change the user password.

### 7.2.2 Constructor & Destructor Documentation

#### 7.2.2.1 ChangePass()

```
es.deusto.client.windows.ChangePass.ChangePass (
    Controller controller )
```

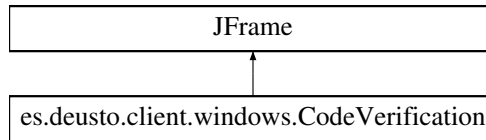
Create the frame.

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

- [src/main/java/es/deusto/client/windows/ChangePass.java](#)

## 7.3 es.deusto.client.windows.CodeVerification Class Reference

Inheritance diagram for es.deusto.client.windows.CodeVerification:



### Public Member Functions

- [CodeVerification](#) (String code, [Controller](#) controller)

*Create the frame.*

### 7.3.1 Constructor & Destructor Documentation

#### 7.3.1.1 CodeVerification()

```
es.deusto.client.windows.CodeVerification.CodeVerification (
    String code,
    Controller controller )
```

Create the frame.

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

- src/main/java/es/deusto/client/windows/[CodeVerification.java](#)

## 7.4 es.deusto.client.controller.Controller Class Reference

This class is the one carrying out all the communication with the server.

## Public Member Functions

- [Controller](#) (String hostname, String port)  
*Constructor.*
- [Controller](#) (WebTarget target)  
*Constructor.*
- [UserInfo](#) [getUser](#) ()
- [OrganizerInfo](#) [getOrganize](#) ()
- void [setOrganizer](#) ([OrganizerInfo](#) org)
- void [setUser](#) ([UserInfo](#) u)
- boolean [attemptNormalLogin](#) (String email, String password)  
*Invokes a normal login attempt call to the server.*
- boolean [attemptNormalLoginOrganizer](#) (String email, String password)  
*Invokes an organizer login attempt call to the server.*
- boolean [attemptNormalLogin](#) (String email, String password, boolean organizer)  
*Invokes a normal login attempt call to the server.*
- boolean [attemptNormalSignup](#) (String email, String password, String name, String city, String country, Array↔  
List< [TopicInfo](#) >interests)  
*Invokes a normal signup attempt call to the server.*
- boolean [attemptOrganizerSignup](#) (String email, String password, String name, String organization)  
*Invokes an organizer signup attempt call to the server.*
- boolean [attemptNormalUpdate](#) ()  
*Invokes a normal user update call to the server.*
- boolean [attemptOrganizerUpdate](#) ()  
*Invokes an organizer update call to the server.*
- boolean [attemptUserDelete](#) ()  
*Invokes a user delete call to the server.*
- boolean [attemptOrganizerDelete](#) ()  
*Invokes an organizer delete call to the server.*
- boolean [createEvent](#) ([EventInfo](#) eventInfo)  
*Invokes an event creation call to the server.*
- boolean [createPost](#) ([EventInfo](#) eventInfo, [PostInfo](#) postInfo)  
*Invokes a post creation call to the server.*
- [LanguageManager](#) [getLanguageManager](#) ()
- ArrayList< [EventInfo](#) > [getRecommendations](#) ()  
*Invokes a recommendations request call to the server.*
- void [attemptPasswordRecovery](#) (String email)  
*Invokes a password recovery call to the server.*
- boolean [attemptEventDelete](#) ([EventInfo](#) info)  
*Invokes a event removal call to the server.*

### 7.4.1 Detailed Description

This class is the one carrying out all the communication with the server.

The GUI components use it as a middle-man in order to retrieve or update information of the server.

This class will contain the [UserInfo](#) property containing the information of the user who has logged in, and will offer it to the components of the GUI.

## 7.4.2 Constructor & Destructor Documentation

### 7.4.2.1 Controller() [1/2]

```
es.deusto.client.controller.Controller.Controller (
    String hostname,
    String port )
```

Constructor.

#### Parameters

<i>hostname</i>	A String object with the path of the server
<i>port</i>	A String object with the port number of the server

### 7.4.2.2 Controller() [2/2]

```
es.deusto.client.controller.Controller.Controller (
    WebTarget target )
```

Constructor.

#### Parameters

<i>target</i>	WebTarget instance pointing to the server (used for testing)
---------------	--

Since

Sprint 2

## 7.4.3 Member Function Documentation

### 7.4.3.1 attemptEventDelete()

```
boolean es.deusto.client.controller.Controller.attemptEventDelete (
    EventInfo info )
```

Invokes a event removal call to the server.

Sends to the server the information of the event, and the server deletes it from the database.

## Parameters

<i>info</i>	event info object
-------------	-------------------

## Returns

## Since

Sprint 3

**7.4.3.2 attemptNormalLogin()** [1/2]

```
boolean es.deusto.client.controller.Controller.attemptNormalLogin (
    String email,
    String password )
```

Invokes a normal login attempt call to the server.

This method is invoked by the login button in the GUI, and it makes a login request to the server with the specified email and password. It is used only for regular users, not organizers.

## Parameters

<i>email</i>	Email of the user taken from the GUI
<i>password</i>	Password taken from the GUI

## Returns

true if the login was successful, otherwise it returns false

## Since

Sprint 1

**7.4.3.3 attemptNormalLogin()** [2/2]

```
boolean es.deusto.client.controller.Controller.attemptNormalLogin (
    String email,
    String password,
    boolean organizer )
```

Invokes a normal login attempt call to the server.

This method is invoked by the login button in the GUI, and it makes a login request to the server with the specified email and password. The class of object must be indicated in the boolean input. This method works as a solution for both organizers and logins

**Parameters**

<i>email</i>	email of the User or Organizer
<i>password</i>	password of the of the UserInfo or OrganizerInfo
<i>organizer</i>	boolean that indicates whether it's a User or Organizer

**Deprecated** beeter to use attemptNormalLogin(email, password) and attemptNormalLoginOrganizer !

**Since**

Sprint 2

**Returns**

boolean if succesful

**7.4.3.4 attemptNormalLoginOrganizer()**

```
boolean es.deusto.client.controller.Controller.attemptNormalLoginOrganizer (
    String email,
    String password )
```

Invokes an organizer login attempt call to the server.

This method is invoked by the login button in the GUI, and it makes a login request to the server with the specified email and password. It is used only for organizers.

**Parameters**

<i>email</i>	email of the Organizer
<i>password</i>	password of the organizer

**Returns**

true if succesful

**Since**

Sprint 2



### 7.4.3.5 attemptNormalSignup()

```
boolean es.deusto.client.controller.Controller.attemptNormalSignup (
    String email,
    String password,
    String name,
    String city,
    String country,
    ArrayList< TopicInfo > interests )
```

Invokes a normal signup attempt call to the server.

This method is invoked by the signup button in the GUI, and makes a signup request to the server with the specified fields (email, password, name and city). It is used only for regular users, not organizers.

#### Parameters

<i>email</i>	Email String taken from the GUI
<i>password</i>	Password String taken from the GUI
<i>name</i>	Name String taken from the GUI
<i>city</i>	City String taken from the GUI
<i>country</i>	Country String taken from the GUI
<i>interests</i>	List of TopicInfo

#### Returns

true if the signup process was successful, otherwise it returns false

#### Since

Sprint 1

### 7.4.3.6 attemptNormalUpdate()

```
boolean es.deusto.client.controller.Controller.attemptNormalUpdate ( )
```

Invokes a normal user update call to the server.

Updates the user stored in the [Controller](#), in order to use this function a user must log in before.

#### Returns

true if the process is successful

#### Since

Sprint 2

#### 7.4.3.7 attemptOrganizerDelete()

```
boolean es.deusto.client.controller.Controller.attemptOrganizerDelete ( )
```

Invokes an organizer delete call to the server.

Deletes the organizer stored in the [Controller](#).

##### Returns

true if the process is successful, false if not

##### Since

Sprint 2

##### Returns

#### 7.4.3.8 attemptOrganizerSignup()

```
boolean es.deusto.client.controller.Controller.attemptOrganizerSignup (
    String email,
    String password,
    String name,
    String organization )
```

Invokes an organizer signup attempt call to the server.

This method is invoked by the signup button in the GUI, and makes a signup request to the server with the specified fields (email, password, name and organization). It is used only for organizers.

##### Parameters

<i>email</i>	Email String taken from the GUI
<i>password</i>	Password String taken from the GUI
<i>name</i>	Name String taken from the GUI
<i>organization</i>	String

##### Since

Sprint 2

##### Returns

true if the signup process was successful, otherwise it returns false

#### 7.4.3.9 attemptOrganizerUpdate()

```
boolean es.deusto.client.controller.Controller.attemptOrganizerUpdate ( )
```

Invokes an organizer update call to the server.

Updates the Organizer stored in the [Controller](#), in order to use this function a Organizer must log in before.

##### Returns

true if the process is successful

##### Since

Sprint 2

#### 7.4.3.10 attemptPasswordRecovery()

```
void es.deusto.client.controller.Controller.attemptPasswordRecovery (
    String email )
```

Invokes a password recovery call to the server.

Sends to the server the email of the user, and the server changes its password and sends it to the email account of the user.

##### Parameters

<i>email</i>	String containing the user's email
--------------	------------------------------------

##### Since

Sprint 3

#### 7.4.3.11 attemptUserDelete()

```
boolean es.deusto.client.controller.Controller.attemptUserDelete ( )
```

Invokes a user delete call to the server.

Deletes the user stored in the [Controller](#).

##### Returns

true if the process is successful, false if not

##### Since

Sprint 2

#### 7.4.3.12 createEvent()

```
boolean es.deusto.client.controller.Controller.createEvent (
    EventInfo eventInfo )
```

Invokes an event creation call to the server.

Sends EventInfo to the server, to use this method the organizer must be logged in.

##### Parameters

<i>eventInfo</i>	EventInfo to be sent to the server
------------------	------------------------------------

##### Returns

true if succesful, false otherwise

##### Since

sprint 2

#### 7.4.3.13 createPost()

```
boolean es.deusto.client.controller.Controller.createPost (
    EventInfo eventInfo,
    PostInfo postInfo )
```

Invokes a post creation call to the server.

Adds a post on the indicated Event and notifies the server.

##### Parameters

<i>eventInfo</i>	EventInfo where the post is going to be held.
<i>postInfo</i>	PostInfo to be added.

##### Returns

true if succesful

##### Since

sprint 2

#### 7.4.3.14 getLanguageManager()

`LanguageManager` es.deusto.client.controller.Controller.getLanguageManager ( )

#### 7.4.3.15 getOrganize()

`OrganizerInfo` es.deusto.client.controller.Controller.getOrganize ( )

#### 7.4.3.16 getRecommendations()

`ArrayList<EventInfo>` es.deusto.client.controller.Controller.getRecommendations ( )

Invokes a recommendations request call to the server.

Sends to the server the information of the logged User in order to get event recommendations

##### Returns

a list of recommended events

##### Since

Sprint 3

#### 7.4.3.17 getUser()

`UserInfo` es.deusto.client.controller.Controller.getUser ( )

#### 7.4.3.18 setOrganizer()

```
void es.deusto.client.controller.Controller.setOrganizer (
    OrganizerInfo org )
```

#### 7.4.3.19 setUser()

```
void es.deusto.client.controller.Controller.setUser (
    UserInfo u )
```

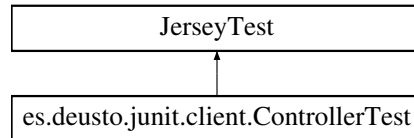
The documentation for this class was generated from the following file:

- src/main/java/es/deusto/client/controller/[Controller.java](#)

## 7.5 es.deusto.junit.client.ControllerTest Class Reference

This classes is the one that tests the controller component to see if it is working as intended.

Inheritance diagram for es.deusto.junit.client.ControllerTest:



### Public Member Functions

- Application [configure](#) ()
- void [setUpChild](#) ()
- void [testCreateDeleteEvent](#) ()
- void [testNormalLogin](#) ()
- void [testOrganizerLogin](#) ()
- void [testNormalSignupAndDelete](#) ()
- void [testOrganizerSignupAndDelete](#) ()
- void [testUserUpdate](#) ()
- void [tearDownChild](#) ()

### Static Public Member Functions

- static void [startUp](#) ()
- static junit.framework.Test [suite](#) ()

### Public Attributes

- ContiPerfRule [rule](#) = new ContiPerfRule()

### 7.5.1 Detailed Description

This classes is the one that tests the controller component to see if it is working as intended.

### 7.5.2 Member Function Documentation

#### 7.5.2.1 configure()

```
Application es.deusto.junit.client.ControllerTest.configure ( )
```

### 7.5.2.2 setUpChild()

```
void es.deusto.junit.client.ControllerTest.setUpChild ( )
```

### 7.5.2.3 startUp()

```
static void es.deusto.junit.client.ControllerTest.startUp ( ) [static]
```

### 7.5.2.4 suite()

```
static junit.framework.Test es.deusto.junit.client.ControllerTest.suite ( ) [static]
```

### 7.5.2.5 tearDownChild()

```
void es.deusto.junit.client.ControllerTest.tearDownChild ( )
```

### 7.5.2.6 testCreateDeleteEvent()

```
void es.deusto.junit.client.ControllerTest.testCreateDeleteEvent ( )
```

### 7.5.2.7 testNormalLogin()

```
void es.deusto.junit.client.ControllerTest.testNormalLogin ( )
```

### 7.5.2.8 testNormalSignupAndDelete()

```
void es.deusto.junit.client.ControllerTest.testNormalSignupAndDelete ( )
```

### 7.5.2.9 testOrganizerLogin()

```
void es.deusto.junit.client.ControllerTest.testOrganizerLogin ( )
```

#### 7.5.2.10 testOrganizerSignupAndDelete()

```
void es.deusto.junit.client.ControllerTest.testOrganizerSignupAndDelete ( )
```

#### 7.5.2.11 testUserUpdate()

```
void es.deusto.junit.client.ControllerTest.testUserUpdate ( )
```

### 7.5.3 Member Data Documentation

#### 7.5.3.1 rule

```
ContiPerfRule es.deusto.junit.client.ControllerTest.rule = new ContiPerfRule()
```

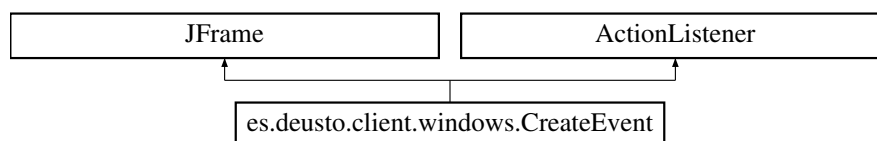
The documentation for this class was generated from the following file:

- [src/test/java/es/deusto/junit/client/ControllerTest.java](#)

## 7.6 es.deusto.client.windows.CreateEvent Class Reference

This class is the one used to create a window that it allows the user to create a new event.

Inheritance diagram for es.deusto.client.windows.CreateEvent:



### Public Member Functions

- [CreateEvent](#) ([Controller](#) controller)
- void [actionPerformed](#) ([ActionEvent](#) e)

#### 7.6.1 Detailed Description

This class is the one used to create a window that it allows the user to create a new event.



## 7.6.2 Constructor & Destructor Documentation

### 7.6.2.1 CreateEvent()

```
es.deusto.client.windows.CreateEvent.CreateEvent (
    Controller controller )
```

## 7.6.3 Member Function Documentation

### 7.6.3.1 actionPerformed()

```
void es.deusto.client.windows.CreateEvent.actionPerformed (
    ActionEvent e )
```

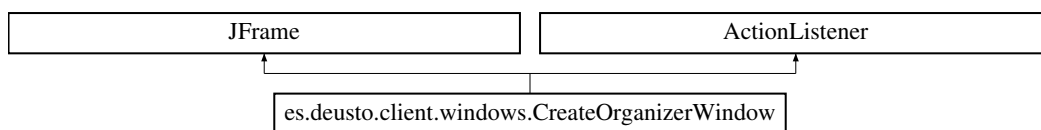
The documentation for this class was generated from the following file:

- src/main/java/es/deusto/client/windows/[CreateEvent.java](#)

## 7.7 es.deusto.client.windows.CreateOrganizerWindow Class Reference

This class is the one used to create a window that allows to create a new organizer.

Inheritance diagram for es.deusto.client.windows.CreateOrganizerWindow:



## Public Member Functions

- [CreateOrganizerWindow](#) ([Controller](#) controller)
- void [actionPerformed](#) (`ActionEvent e`)

### 7.7.1 Detailed Description

This class is the one used to create a window that allows to create a new organizer.

## 7.7.2 Constructor & Destructor Documentation

### 7.7.2.1 CreateOrganizerWindow()

```
es.deusto.client.windows.CreateOrganizerWindow.CreateOrganizerWindow (
    Controller controller )
```

## 7.7.3 Member Function Documentation

### 7.7.3.1 actionPerformed()

```
void es.deusto.client.windows.CreateOrganizerWindow.actionPerformed (
    ActionEvent e )
```

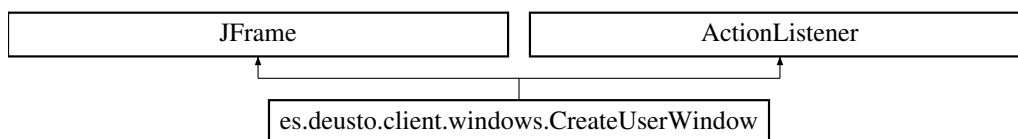
The documentation for this class was generated from the following file:

- [src/main/java/es/deusto/client/windows/CreateOrganizerWindow.java](#)

## 7.8 es.deusto.client.windows.CreateUserWindow Class Reference

This class is the one used to create a window that allows to create a new user.

Inheritance diagram for es.deusto.client.windows.CreateUserWindow:



### Public Member Functions

- [CreateUserWindow](#) (Controller controller)
- void [actionPerformed](#) (ActionEvent e)

### 7.8.1 Detailed Description

This class is the one used to create a window that allows to create a new user.

## 7.8.2 Constructor & Destructor Documentation

### 7.8.2.1 CreateUserWindow()

```
es.deusto.client.windows.CreateUserWindow.CreateUserWindow (
    Controller controller )
```

## 7.8.3 Member Function Documentation

### 7.8.3.1 actionPerformed()

```
void es.deusto.client.windows.CreateUserWindow.actionPerformed (
    ActionEvent e )
```

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

- [src/main/java/es/deusto/client/windows/CreateUserWindow.java](#)

## 7.9 es.deusto.server.dao.DAOFactory Class Reference

This class is used to create a DAO factory that is used by all of the other DAO classes, to create them or to close them.

### Public Member Functions

- [UserDAO](#) [createUserDAO](#) ()
- [OrganizerDAO](#) [createOrganizerDAO](#) ()
- [EventDAO](#) [createEventDAO](#) ()
- [TopicDAO](#) [createTopicDAO](#) ()
- [PostDAO](#) [createPostDAO](#) ()
- void [closeDAO](#) ([PostDAO](#) dao)
- void [closeDAO](#) ([TopicDAO](#) dao)
- void [closeDAO](#) ([EventDAO](#) dao)
- void [closeDAO](#) ([OrganizerDAO](#) dao)
- void [closeDAO](#) ([UserDAO](#) dao)

### Static Public Member Functions

- static [DAOFactory](#) [getInstance](#) ()

## 7.9.1 Detailed Description

This class is used to create a DAO factory that is used by all of the other DAO classes, to create them or to close them.

## 7.9.2 Member Function Documentation

### 7.9.2.1 closeDAO() [1/5]

```
void es.deusto.server.dao.DAOFactory.closeDAO (
    EventDAO dao )
```

### 7.9.2.2 closeDAO() [2/5]

```
void es.deusto.server.dao.DAOFactory.closeDAO (
    OrganizerDAO dao )
```

### 7.9.2.3 closeDAO() [3/5]

```
void es.deusto.server.dao.DAOFactory.closeDAO (
    PostDAO dao )
```

### 7.9.2.4 closeDAO() [4/5]

```
void es.deusto.server.dao.DAOFactory.closeDAO (
    TopicDAO dao )
```

### 7.9.2.5 closeDAO() [5/5]

```
void es.deusto.server.dao.DAOFactory.closeDAO (
    UserDAO dao )
```

### 7.9.2.6 createEventDAO()

```
EventDAO es.deusto.server.dao.DAOFactory.createEventDAO ( )
```

### 7.9.2.7 createOrganizerDAO()

```
OrganizerDAO es.deusto.server.dao.DAOFactory.createOrganizerDAO ( )
```

### 7.9.2.8 createPostDAO()

```
PostDAO es.deusto.server.dao.DAOFactory.createPostDAO ( )
```

### 7.9.2.9 createTopicDAO()

```
TopicDAO es.deusto.server.dao.DAOFactory.createTopicDAO ( )
```

### 7.9.2.10 createUserDAO()

```
UserDAO es.deusto.server.dao.DAOFactory.createUserDAO ( )
```

### 7.9.2.11 getInstance()

```
static DAOFactory es.deusto.server.dao.DAOFactory.getInstance ( ) [static]
```

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

- src/main/java/es/deusto/server/dao/DAOFactory.java

## 7.10 es.deusto.server.data.Event Class Reference

This classes is the one that defines an event and it is used to create one.

## Public Member Functions

- [Event](#) ()
- [Event](#) ([EventInfo](#) info)
- String [getName](#) ()
- void [setName](#) (String name)
- String [getDescription](#) ()
- void [setDescription](#) (String desc)
- [Topic](#) [getTopic](#) ()
- void [setTopic](#) ([Topic](#) topic)
- [Organizer](#) [getOrganizer](#) ()
- void [setOrganizer](#) ([Organizer](#) organizer)
- String [getCity](#) ()
- void [setCity](#) (String city)
- String [getCountry](#) ()
- void [setCountry](#) (String country)
- Date [getDate](#) ()
- void [setDate](#) (Date date)
- int [getInterested](#) ()
- void [setInterested](#) (int interested)
- void [addInterested](#) ()
- void [reduceInterested](#) ()
- ArrayList< [Post](#) > [getPosts](#) ()
- void [setPosts](#) (ArrayList< [Post](#) > posts)
- String [toString](#) ()

### 7.10.1 Detailed Description

This classes is the one that defines an event and it is used to create one.

### 7.10.2 Constructor & Destructor Documentation

#### 7.10.2.1 [Event\(\)](#) [1/2]

```
es.deusto.server.data.Event.Event ( )
```

#### 7.10.2.2 [Event\(\)](#) [2/2]

```
es.deusto.server.data.Event.Event (
    EventInfo info )
```

### 7.10.3 Member Function Documentation

#### 7.10.3.1 addInterested()

```
void es.deusto.server.data.Event.addInterested ( )
```

#### 7.10.3.2 getCity()

```
String es.deusto.server.data.Event.getCity ( )
```

#### 7.10.3.3 getCountry()

```
String es.deusto.server.data.Event.getCountry ( )
```

#### 7.10.3.4 getDate()

```
Date es.deusto.server.data.Event.getDate ( )
```

#### 7.10.3.5 getDescription()

```
String es.deusto.server.data.Event.getDescription ( )
```

#### 7.10.3.6 getInterested()

```
int es.deusto.server.data.Event.getInterested ( )
```

#### 7.10.3.7 getName()

```
String es.deusto.server.data.Event.getName ( )
```

#### 7.10.3.8 getOrganizer()

```
Organizer es.deusto.server.data.Event.getOrganizer ( )
```

#### 7.10.3.9 getPosts()

```
ArrayList<Post> es.deusto.server.data.Event.getPosts ( )
```

#### 7.10.3.10 getTopic()

```
Topic es.deusto.server.data.Event.getTopic ( )
```

#### 7.10.3.11 reduceInterested()

```
void es.deusto.server.data.Event.reduceInterested ( )
```

#### 7.10.3.12 setCity()

```
void es.deusto.server.data.Event.setCity (
    String city )
```

#### 7.10.3.13 setCountry()

```
void es.deusto.server.data.Event.setCountry (
    String country )
```

#### 7.10.3.14 setDate()

```
void es.deusto.server.data.Event.setDate (
    Date date )
```



### 7.10.3.15 setDescription()

```
void es.deusto.server.data.Event.setDescription (
    String desc )
```

### 7.10.3.16 setInterested()

```
void es.deusto.server.data.Event.setInterested (
    int interested )
```

### 7.10.3.17 setName()

```
void es.deusto.server.data.Event.setName (
    String name )
```

### 7.10.3.18 setOrganizer()

```
void es.deusto.server.data.Event.setOrganizer (
    Organizer organizer )
```

### 7.10.3.19 setPosts()

```
void es.deusto.server.data.Event.setPosts (
    ArrayList< Post > posts )
```

### 7.10.3.20 setTopic()

```
void es.deusto.server.data.Event.setTopic (
    Topic topic )
```

### 7.10.3.21 toString()

```
String es.deusto.server.data.Event.toString ( )
```

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

- src/main/java/es/deusto/server/data/[Event.java](#)

## 7.11 es.deusto.server.dao.EventDAO Class Reference

This classes is the one used to create an event DAO.

### Public Member Functions

- ArrayList< [Event](#) > [getEvents](#) (String name)  
*Retrieves List of events from the database given the full name or part of it.*
- ArrayList< [Event](#) > [getEventsbyUser](#) ([User](#) user)  
*Retrieves List of events from the database that match the user's interests and location.*
- ArrayList< [Event](#) > [getEventsByOrganizer](#) ([Organizer](#) organizer)  
*Retrieves List of the events of a specific Organizer.*
- ArrayList< [Event](#) > [getEvents](#) ()  
*Retrieves a list of all the events from the database.*
- void [storeEvent](#) ([Event](#) event)  
*Stores a event in the database.*
- void [updateEvent](#) ([Event](#) event)  
*Updates an event.*
- void [deleteEvent](#) ([Event](#) event)  
*Deletes an event from the database.*

### Protected Member Functions

- [EventDAO](#) (PersistenceManager pm)
- PersistenceManager [getPersistenceManager](#) ()
- void [close](#) ()

#### 7.11.1 Detailed Description

This classes is the one used to create an event DAO.

#### 7.11.2 Constructor & Destructor Documentation

##### 7.11.2.1 EventDAO()

```
es.deusto.server.dao.EventDAO.EventDAO (  
    PersistenceManager pm ) [protected]
```

#### 7.11.3 Member Function Documentation

### 7.11.3.1 close()

```
void es.deusto.server.dao.EventDAO.close ( ) [protected]
```

### 7.11.3.2 deleteEvent()

```
void es.deusto.server.dao.EventDAO.deleteEvent (
    Event event )
```

deletes an event from the database.

#### Parameters

<i>event</i>	event to be deleted
--------------	---------------------

### 7.11.3.3 getEvents() [1/2]

```
ArrayList<Event> es.deusto.server.dao.EventDAO.getEvents ( )
```

Retrieves a list of all the events from the database.

### 7.11.3.4 getEvents() [2/2]

```
ArrayList<Event> es.deusto.server.dao.EventDAO.getEvents (
    String name )
```

Retrieves List of events from the database given the full name or part of it.

#### Parameters

<i>name</i>	name of the event
-------------	-------------------

#### Returns

list of events from the database

### 7.11.3.5 getEventsByOrganizer()

```
ArrayList<Event> es.deusto.server.dao.EventDAO.getEventsByOrganizer (
    Organizer organizer )
```

Retrieves List of the events of a specific Organizer.

**Parameters**

<i>organizer</i>	
------------------	--

**Returns**

list of events from the database

**7.11.3.6 getEventsbyUser()**

```
ArrayList<Event> es.deusto.server.dao.EventDAO.getEventsbyUser (
    User user )
```

Retrieves List of events from the database that match the user's interests and location.

this is use to create recomendation lists.

**Parameters**

<i>user</i>	User
-------------	------

**Returns**

list of events from the database

**7.11.3.7 getPersistenceManager()**

```
PersistenceManager es.deusto.server.dao.EventDAO.getPersistenceManager ( ) [protected]
```

**7.11.3.8 storeEvent()**

```
void es.deusto.server.dao.EventDAO.storeEvent (
    Event event )
```

Stores a event in the database.

**Parameters**

<i>event</i>	event to be submitted
--------------	-----------------------

### 7.11.3.9 updateEvent()

```
void es.deusto.server.dao.EventDAO.updateEvent (
    Event event )
```

Updates an event.

#### Parameters

<i>event</i>	
--------------	--

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

- src/main/java/es/deusto/server/dao/[EventDAO.java](#)

## 7.12 es.deusto.serialization.EventInfo Class Reference

DTO equivalent class of the Event class from the server.

### Public Member Functions

- [EventInfo](#) ()
- [EventInfo](#) ([Event](#) e)
- String [getName](#) ()
- void [setName](#) (String name)
- String [getDescription](#) ()
- void [setDescription](#) (String description)
- [TopicInfo](#) [getTopic](#) ()
- void [setTopic](#) ([TopicInfo](#) topic)
- String [getOrganizerEmail](#) ()
- void [setOrganizerEmail](#) (String organizerEmail)
- ArrayList< [PostInfo](#) > [getPosts](#) ()
- void [setPosts](#) (ArrayList< [PostInfo](#) > posts)
- int [getInterested](#) ()
- void [setInterested](#) (int interested)
- void [addInterested](#) ()
- void [reduceInterested](#) ()
- String [getCity](#) ()
- void [setCity](#) (String city)
- String [getCountry](#) ()
- void [setCountry](#) (String country)
- Date [getDate](#) ()
- void [setDate](#) (Date date)

### 7.12.1 Detailed Description

DTO equivalent class of the Event class from the server.

This class is used to transport all the relevant information of an event through the network.

Since

Sprint 1

## 7.12.2 Constructor & Destructor Documentation

### 7.12.2.1 EventInfo() [1/2]

```
es.deusto.serialization.EventInfo.EventInfo ( )
```

### 7.12.2.2 EventInfo() [2/2]

```
es.deusto.serialization.EventInfo.EventInfo (
    Event e )
```

## 7.12.3 Member Function Documentation

### 7.12.3.1 addInterested()

```
void es.deusto.serialization.EventInfo.addInterested ( )
```

### 7.12.3.2 getCity()

```
String es.deusto.serialization.EventInfo.getCity ( )
```

### 7.12.3.3 getCountry()

```
String es.deusto.serialization.EventInfo.getCountry ( )
```

### 7.12.3.4 getDate()

```
Date es.deusto.serialization.EventInfo.getDate ( )
```

### 7.12.3.5 getDescription()

```
String es.deusto.serialization.EventInfo.getDescription ( )
```

### 7.12.3.6 getInterested()

```
int es.deusto.serialization.EventInfo.getInterested ( )
```

### 7.12.3.7 getName()

```
String es.deusto.serialization.EventInfo.getName ( )
```

### 7.12.3.8 getOrganizerEmail()

```
String es.deusto.serialization.EventInfo.getOrganizerEmail ( )
```

### 7.12.3.9 getPosts()

```
ArrayList<PostInfo> es.deusto.serialization.EventInfo.getPosts ( )
```

### 7.12.3.10 getTopic()

```
TopicInfo es.deusto.serialization.EventInfo.getTopic ( )
```

### 7.12.3.11 reduceInterested()

```
void es.deusto.serialization.EventInfo.reduceInterested ( )
```

### 7.12.3.12 setCity()

```
void es.deusto.serialization.EventInfo.setCity (
    String city )
```

**7.12.3.13 setCountry()**

```
void es.deusto.serialization.EventInfo.setCountry (
    String country )
```

**7.12.3.14 setDate()**

```
void es.deusto.serialization.EventInfo.setDate (
    Date date )
```

**7.12.3.15 setDescription()**

```
void es.deusto.serialization.EventInfo.setDescription (
    String description )
```

**7.12.3.16 setInterested()**

```
void es.deusto.serialization.EventInfo.setInterested (
    int interested )
```

**7.12.3.17 setName()**

```
void es.deusto.serialization.EventInfo.setName (
    String name )
```

**7.12.3.18 setOrganizerEmail()**

```
void es.deusto.serialization.EventInfo.setOrganizerEmail (
    String organizerEmail )
```

**7.12.3.19 setPosts()**

```
void es.deusto.serialization.EventInfo.setPosts (
    ArrayList< PostInfo > posts )
```



### 7.12.3.20 setTopic()

```
void es.deusto.serialization.EventInfo.setTopic (
    TopicInfo topic )
```

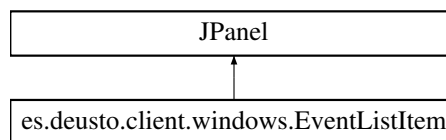
The documentation for this class was generated from the following file:

- src/main/java/es/deusto/serialization/EventInfo.java

## 7.13 es.deusto.client.windows.EventListItem Class Reference

This class is the one used to create a window that allows the user to have an event list.

Inheritance diagram for es.deusto.client.windows.EventListItem:



### Public Member Functions

- `EventListItem` (`Controller` controller, `EventInfo` event, `JFrame` parent)

### 7.13.1 Detailed Description

This class is the one used to create a window that allows the user to have an event list.

### 7.13.2 Constructor & Destructor Documentation

#### 7.13.2.1 EventListItem()

```
es.deusto.client.windows.EventListItem.EventListItem (
    Controller controller,
    EventInfo event,
    JFrame parent )
```

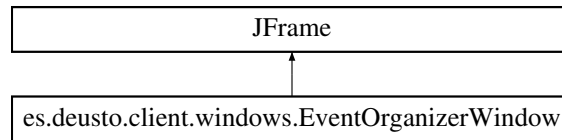
The documentation for this class was generated from the following file:

- src/main/java/es/deusto/client/windows/EventListItem.java

## 7.14 es.deusto.client.windows.EventOrganizerWindow Class Reference

This class is the one used to create a window that allows to an organizer to change or administrate one of his events.

Inheritance diagram for es.deusto.client.windows.EventOrganizerWindow:



### Public Member Functions

- [EventOrganizerWindow](#) ([Controller](#) controller)

#### 7.14.1 Detailed Description

This class is the one used to create a window that allows to an organizer to change or administrate one of his events.

#### 7.14.2 Constructor & Destructor Documentation

##### 7.14.2.1 EventOrganizerWindow()

```
es.deusto.client.windows.EventOrganizerWindow.EventOrganizerWindow (
    Controller controller )
```

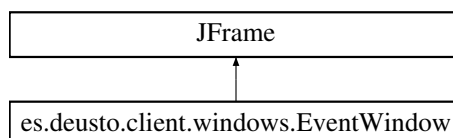
The documentation for this class was generated from the following file:

- src/main/java/es/deusto/client/windows/[EventOrganizerWindow.java](#)

## 7.15 es.deusto.client.windows.EventWindow Class Reference

This class is the one used to create a window that allows to display all the information of one event.

Inheritance diagram for es.deusto.client.windows.EventWindow:



## Public Member Functions

- [EventWindow](#) ([Controller](#) controller, [EventInfo](#) e)

### 7.15.1 Detailed Description

This class is the one used to create a window that allows to display all the information of one event.

### 7.15.2 Constructor & Destructor Documentation

#### 7.15.2.1 EventWindow()

```
es.deusto.client.windows.EventWindow.EventWindow (
    Controller controller,
    EventInfo e )
```

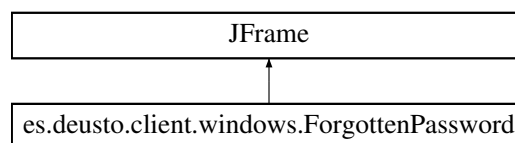
The documentation for this class was generated from the following file:

- src/main/java/es/deusto/client/windows/[EventWindow.java](#)

## 7.16 es.deusto.client.windows.ForgottenPassword Class Reference

This class is the one used to create a window that allows a user to log in when he forgets the password.

Inheritance diagram for es.deusto.client.windows.ForgottenPassword:



## Public Member Functions

- [ForgottenPassword](#) ([Controller](#) controller)  
*Create the frame.*

### 7.16.1 Detailed Description

This class is the one used to create a window that allows a user to log in when he forgets the password.

## 7.16.2 Constructor & Destructor Documentation

### 7.16.2.1 ForgottenPassword()

```
es.deusto.client.windows.ForgottenPassword.ForgottenPassword (
    Controller controller )
```

Create the frame.

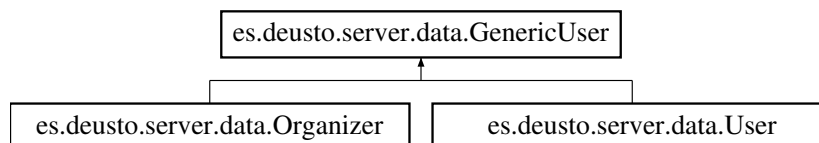
The documentation for this class was generated from the following file:

- [src/main/java/es/deusto/client/windows/ForgottenPassword.java](#)

## 7.17 es.deusto.server.data.GenericUser Class Reference

This classes is the father class for normal user and organizer.

Inheritance diagram for es.deusto.server.data.GenericUser:



### Public Member Functions

- String [getName](#) ()
- void [setName](#) (String name)
- String [getEmail](#) ()
- void [setEmail](#) (String email)
- String [getPassword](#) ()
- void [setPassword](#) (String password)
- String [toString](#) ()

### 7.17.1 Detailed Description

This classes is the father class for normal user and organizer.

Both of them are users so they inherit from this class.

### 7.17.2 Member Function Documentation

### 7.17.2.1 getEmail()

```
String es.deusto.server.data.GenericUser.getEmail ( )
```

### 7.17.2.2 getName()

```
String es.deusto.server.data.GenericUser.getName ( )
```

### 7.17.2.3 getPassword()

```
String es.deusto.server.data.GenericUser.getPassword ( )
```

### 7.17.2.4 setEmail()

```
void es.deusto.server.data.GenericUser.setEmail (
    String email )
```

### 7.17.2.5 setName()

```
void es.deusto.server.data.GenericUser.setName (
    String name )
```

### 7.17.2.6 setPassword()

```
void es.deusto.server.data.GenericUser.setPassword (
    String password )
```

### 7.17.2.7 toString()

```
String es.deusto.server.data.GenericUser.toString ( )
```

Reimplemented in [es.deusto.server.data.User](#), and [es.deusto.server.data.Organizer](#).

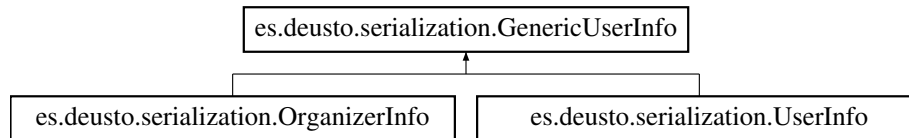
The documentation for this class was generated from the following file:

- [src/main/java/es/deusto/server/data/GenericUser.java](#)

## 7.18 es.deusto.serialization.GenericUserInfo Class Reference

This class works as a Tag for the Classes related to the User and Organizer.

Inheritance diagram for es.deusto.serialization.GenericUserInfo:



### 7.18.1 Detailed Description

This class works as a Tag for the Classes related to the User and Organizer.

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

- [src/main/java/es/deusto/serialization/GenericUserInfo.java](#)

## 7.19 es.deusto.client.windows.LanguageManager Class Reference

This class is used to manage the different languages supported by the application.

### Public Member Functions

- [LanguageManager](#) ()  
*Constructor.*
- [LanguageManager](#) (String language)  
*Creates an instance of the class and sets the current language to the specified one.*
- String [getString](#) (String key)  
*Retrieves the translation to the current language of the String identified by the key.*
- String [getLanguage](#) ()
- void [setLanguage](#) (String language)  
*Sets the current language.*

### 7.19.1 Detailed Description

This class is used to manage the different languages supported by the application.

It keeps track of the current language, and it allows the users to change it and to get the specified string in the current language.

### 7.19.2 Constructor & Destructor Documentation

### 7.19.2.1 LanguageManager() [1/2]

```
es.deusto.client.windows.LanguageManager.LanguageManager ( )
```

Constructor.

Loads the default system language.

### 7.19.2.2 LanguageManager() [2/2]

```
es.deusto.client.windows.LanguageManager.LanguageManager (
    String language )
```

Creates an instance of the class and sets the current language to the specified one.

#### Parameters

<i>language</i>	The String code of the desired language
-----------------	---

## 7.19.3 Member Function Documentation

### 7.19.3.1 getLanguage()

```
String es.deusto.client.windows.LanguageManager.getLanguage ( )
```

### 7.19.3.2 getString()

```
String es.deusto.client.windows.LanguageManager.getString (
    String key )
```

Retrieves the translation to the current language of the String identified by the key.

#### Parameters

<i>key</i>	Identifier of the desired String
------------	----------------------------------

#### Returns

translation Translation of the specified String to the current language

### 7.19.3.3 setLanguage()

```
void es.deusto.client.windows.LanguageManager.setLanguage (
    String language )
```

Sets the current language.

#### Parameters

<i>language</i>	The String code of the desired language
-----------------	---

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

- [src/main/java/es/deusto/client/windows/LanguageManager.java](#)

## 7.20 es.deusto.serialization.LoginAttempt Class Reference

This class encapsulates all the information required when a user (Either normal User or Organizer) logs in.

### Public Member Functions

- [LoginAttempt](#) ()
- [LoginAttempt](#) (String email, String password, boolean organizer)  
*Constructor of a Log in attempt.*
- String [getEmail](#) ()
- void [setEmail](#) (String email)
- String [getPassword](#) ()
- void [setPassword](#) (String password)
- boolean [isOrganizer](#) ()
- void [setOrganizer](#) (boolean organizer)
- String [toString](#) ()

### 7.20.1 Detailed Description

This class encapsulates all the information required when a user (Either normal User or Organizer) logs in.

Since

Sprint 1

### 7.20.2 Constructor & Destructor Documentation



### 7.20.2.1 LoginAttempt() [1/2]

```
es.deusto.serialization.LoginAttempt.LoginAttempt ( )
```

### 7.20.2.2 LoginAttempt() [2/2]

```
es.deusto.serialization.LoginAttempt.LoginAttempt (
    String email,
    String password,
    boolean organizer )
```

Constructor of a Log in attempt.

Encapsulates the information returned by the server when a user (Either normal User or Organizer) Logs in.

#### Parameters

<i>email</i>	email of the user that will log.
<i>password</i>	password of the account.
<i>organizer</i>	boolean parameter that states if this user is an organizer or not.

## 7.20.3 Member Function Documentation

### 7.20.3.1 getEmail()

```
String es.deusto.serialization.LoginAttempt.getEmail ( )
```

### 7.20.3.2 getPassword()

```
String es.deusto.serialization.LoginAttempt.getPassword ( )
```

### 7.20.3.3 isOrganizer()

```
boolean es.deusto.serialization.LoginAttempt.isOrganizer ( )
```

#### 7.20.3.4 setEmail()

```
void es.deusto.serialization.LoginAttempt.setEmail (
    String email )
```

#### 7.20.3.5 setOrganizer()

```
void es.deusto.serialization.LoginAttempt.setOrganizer (
    boolean organizer )
```

#### 7.20.3.6 setPassword()

```
void es.deusto.serialization.LoginAttempt.setPassword (
    String password )
```

#### 7.20.3.7 toString()

```
String es.deusto.serialization.LoginAttempt.toString ( )
```

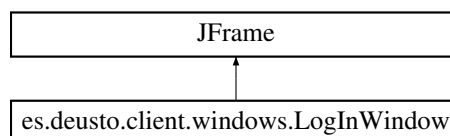
The documentation for this class was generated from the following file:

- [src/main/java/es/deusto/serialization/LoginAttempt.java](#)

## 7.21 es.deusto.client.windows.LogInWindow Class Reference

This class is the one used to create a window that allows a user/organizer to log in to the application.

Inheritance diagram for es.deusto.client.windows.LogInWindow:



### Public Member Functions

- [LogInWindow](#) ([Controller](#) controller)

## Public Attributes

- JTextField [id](#)
- JButton [bCreate](#)
- JCheckBox [cBoxAdmin](#)

### 7.21.1 Detailed Description

This class is the one used to create a window that allows a user/organizer to log in to the application.

### 7.21.2 Constructor & Destructor Documentation

#### 7.21.2.1 LoginWindow()

```
es.deusto.client.windows.LoginWindow.LoginWindow (
    Controller controller )
```

### 7.21.3 Member Data Documentation

#### 7.21.3.1 bCreate

```
JButton es.deusto.client.windows.LoginWindow.bCreate
```

#### 7.21.3.2 cBoxAdmin

```
JCheckBox es.deusto.client.windows.LoginWindow.cBoxAdmin
```

#### 7.21.3.3 id

```
JTextField es.deusto.client.windows.LoginWindow.id
```

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

- [src/main/java/es/deusto/client/windows/LoginWindow.java](#)

## 7.22 es.deusto.client.main.Main Class Reference

This class is the one that starts the application.

### Static Public Member Functions

- static void [main](#) (String[] args) throws ClassNotFoundException

#### 7.22.1 Detailed Description

This class is the one that starts the application.

It creates a new controller for the user to connect to the server and opens a new LoginWindow where the user can start using the application by login in or creating a new user.

#### 7.22.2 Member Function Documentation

##### 7.22.2.1 main()

```
static void es.deusto.client.main.Main.main (  
    String[] args ) throws ClassNotFoundException [static]
```

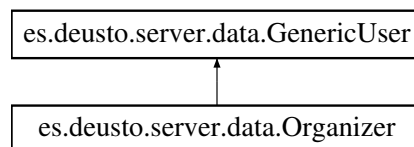
The documentation for this class was generated from the following file:

- src/main/java/es/deusto/client/main/[Main.java](#)

## 7.23 es.deusto.server.data.Organizer Class Reference

This classes is the one that defines an organizer and it is used to create one.

Inheritance diagram for es.deusto.server.data.Organizer:



### Public Member Functions

- [Organizer](#) ()
- String [getOrganization](#) ()
- void [setOrganization](#) (String organization)
- ArrayList< [Event](#) > [getCreatedEvents](#) ()
- void [setCreatedEvents](#) (ArrayList< [Event](#) > createdEvents)
- void [addCreatedEvent](#) ([Event](#) e)
- String [toString](#) ()

### 7.23.1 Detailed Description

This classes is the one that defines an organizer and it is used to create one.

### 7.23.2 Constructor & Destructor Documentation

#### 7.23.2.1 Organizer()

```
es.deusto.server.data.Organizer.Organizer ( )
```

### 7.23.3 Member Function Documentation

#### 7.23.3.1 addCreatedEvent()

```
void es.deusto.server.data.Organizer.addCreatedEvent (
    Event e )
```

#### 7.23.3.2 getCreatedEvents()

```
ArrayList<Event> es.deusto.server.data.Organizer.getCreatedEvents ( )
```

#### 7.23.3.3 getOrganization()

```
String es.deusto.server.data.Organizer.getOrganization ( )
```

#### 7.23.3.4 setCreatedEvents()

```
void es.deusto.server.data.Organizer.setCreatedEvents (
    ArrayList< Event > createdEvents )
```

## Parameters

<i>createdEvents</i>	the createdEvents to set
----------------------	--------------------------

### 7.23.3.5 setOrganization()

```
void es.deusto.server.data.Organizer.setOrganization (
    String organization )
```

### 7.23.3.6 toString()

```
String es.deusto.server.data.Organizer.toString ( )
```

Reimplemented from [es.deusto.server.data.GenericUser](#).

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

- [src/main/java/es/deusto/server/data/Organizer.java](#)

## 7.24 es.deusto.server.dao.OrganizerDAO Class Reference

This classes is the one used to create an organizer DAO.

### Public Member Functions

- [Organizer](#) [getOrganizer](#) (String email)
- void [storeOrganizer](#) ([Organizer](#) organizer)
- void [updateOrganizer](#) ([Organizer](#) organizer)
- void [deleteOrganizer](#) (String email)

### Protected Member Functions

- [OrganizerDAO](#) (PersistenceManager pm)
- PersistenceManager [getPersistenceManager](#) ()
- void [close](#) ()

### 7.24.1 Detailed Description

This classes is the one used to create an organizer DAO.

## 7.24.2 Constructor & Destructor Documentation

### 7.24.2.1 OrganizerDAO()

```
es.deusto.server.dao.OrganizerDAO.OrganizerDAO (
    PersistenceManager pm ) [protected]
```

## 7.24.3 Member Function Documentation

### 7.24.3.1 close()

```
void es.deusto.server.dao.OrganizerDAO.close ( ) [protected]
```

### 7.24.3.2 deleteOrganizer()

```
void es.deusto.server.dao.OrganizerDAO.deleteOrganizer (
    String email )
```

### 7.24.3.3 getOrganizer()

```
Organizer es.deusto.server.dao.OrganizerDAO.getOrganizer (
    String email )
```

### 7.24.3.4 getPersistenceManager()

```
PersistenceManager es.deusto.server.dao.OrganizerDAO.getPersistenceManager ( ) [protected]
```

### 7.24.3.5 storeOrganizer()

```
void es.deusto.server.dao.OrganizerDAO.storeOrganizer (
    Organizer organizer )
```

### 7.24.3.6 updateOrganizer()

```
void es.deusto.server.dao.OrganizerDAO.updateOrganizer (
    Organizer organizer )
```

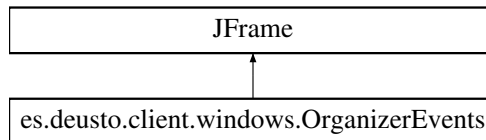
The documentation for this class was generated from the following file:

- [src/main/java/es/deusto/server/dao/OrganizerDAO.java](#)

## 7.25 es.deusto.client.windows.OrganizerEvents Class Reference

This class is the one used to create a window that allows an organizer to have all of his events.

Inheritance diagram for es.deusto.client.windows.OrganizerEvents:



### Public Member Functions

- [OrganizerEvents](#) ([Controller](#) controller)

### 7.25.1 Detailed Description

This class is the one used to create a window that allows an organizer to have all of his events.

### 7.25.2 Constructor & Destructor Documentation

#### 7.25.2.1 OrganizerEvents()

```
es.deusto.client.windows.OrganizerEvents.OrganizerEvents (
    Controller controller )
```

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

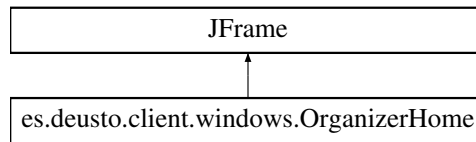
- [src/main/java/es/deusto/client/windows/OrganizerEvents.java](#)



## 7.26 es.deusto.client.windows.OrganizerHome Class Reference

This class is the one used to create a window that allows an organizer to have a home window where he can access to all of the functionalities of the application.

Inheritance diagram for es.deusto.client.windows.OrganizerHome:



### Public Member Functions

- [OrganizerHome](#) ([Controller](#) controller)

#### 7.26.1 Detailed Description

This class is the one used to create a window that allows an organizer to have a home window where he can access to all of the functionalities of the application.

#### 7.26.2 Constructor & Destructor Documentation

##### 7.26.2.1 OrganizerHome()

```
es.deusto.client.windows.OrganizerHome.OrganizerHome (
    Controller controller )
```

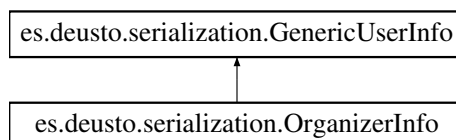
The documentation for this class was generated from the following file:

- [src/main/java/es/deusto/client/windows/OrganizerHome.java](#)

## 7.27 es.deusto.serialization.OrganizerInfo Class Reference

DTO equivalent class of the Organizer class from the server.

Inheritance diagram for es.deusto.serialization.OrganizerInfo:



## Public Member Functions

- [OrganizerInfo](#) ()
- [OrganizerInfo](#) ([Organizer](#) organizer)
- String [getName](#) ()
- void [setName](#) (String name)
- String [getEmail](#) ()
- void [setEmail](#) (String email)
- String [getOrganization](#) ()
- void [setOrganization](#) (String organization)
- ArrayList< [EventInfo](#) > [getCreatedEvents](#) ()
- void [setCreatedEvents](#) (ArrayList< [EventInfo](#) > createdEvents)

### 7.27.1 Detailed Description

DTO equivalent class of the Organizer class from the server.

This class is used to transport all the relevant information of an organizer through the network.

Since

Sprint 1

### 7.27.2 Constructor & Destructor Documentation

#### 7.27.2.1 [OrganizerInfo](#)() [1/2]

```
es.deusto.serialization.OrganizerInfo.OrganizerInfo ( )
```

#### 7.27.2.2 [OrganizerInfo](#)() [2/2]

```
es.deusto.serialization.OrganizerInfo.OrganizerInfo (
    Organizer organizer )
```

### 7.27.3 Member Function Documentation

#### 7.27.3.1 [getCreatedEvents](#)()

```
ArrayList<EventInfo> es.deusto.serialization.OrganizerInfo.getCreatedEvents ( )
```

### 7.27.3.2 getEmail()

```
String es.deusto.serialization.OrganizerInfo.getEmail ( )
```

### 7.27.3.3 getName()

```
String es.deusto.serialization.OrganizerInfo.getName ( )
```

### 7.27.3.4 getOrganization()

```
String es.deusto.serialization.OrganizerInfo.getOrganization ( )
```

### 7.27.3.5 setCreatedEvents()

```
void es.deusto.serialization.OrganizerInfo.setCreatedEvents (
    ArrayList< EventInfo > createdEvents )
```

### 7.27.3.6 setEmail()

```
void es.deusto.serialization.OrganizerInfo.setEmail (
    String email )
```

### 7.27.3.7 setName()

```
void es.deusto.serialization.OrganizerInfo.setName (
    String name )
```

### 7.27.3.8 setOrganization()

```
void es.deusto.serialization.OrganizerInfo.setOrganization (
    String organization )
```

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

- [src/main/java/es/deusto/serialization/OrganizerInfo.java](#)

## 7.28 es.deusto.server.data.Post Class Reference

This classes is the one that defines a post and it is used to create one.

### Public Member Functions

- [Post](#) ()
- [Post](#) ([PostInfo](#) info)
- String [getTitle](#) ()
- void [setTitle](#) (String title)
- String [getDescription](#) ()
- void [setDescription](#) (String description)
- Date [getDate](#) ()
- void [setDate](#) (Date date)
- String [getEventName](#) ()
- void [setEventName](#) (String eventName)
- String [getOrganizerEmail](#) ()
- void [setOrganizerEmail](#) (String organizerEmail)

### 7.28.1 Detailed Description

This classes is the one that defines a post and it is used to create one.

### 7.28.2 Constructor & Destructor Documentation

#### 7.28.2.1 [Post\(\)](#) [1/2]

```
es.deusto.server.data.Post.Post ( )
```

#### 7.28.2.2 [Post\(\)](#) [2/2]

```
es.deusto.server.data.Post.Post (
    PostInfo info )
```

### 7.28.3 Member Function Documentation

### 7.28.3.1 getDate()

```
Date es.deusto.server.data.Post.getDate ( )
```

### 7.28.3.2 getDescription()

```
String es.deusto.server.data.Post.getDescription ( )
```

### 7.28.3.3 getEventName()

```
String es.deusto.server.data.Post.getEventName ( )
```

### 7.28.3.4 getOrganizerEmail()

```
String es.deusto.server.data.Post.getOrganizerEmail ( )
```

### 7.28.3.5 getTitle()

```
String es.deusto.server.data.Post.getTitle ( )
```

### 7.28.3.6 setDate()

```
void es.deusto.server.data.Post.setDate (
    Date date )
```

### 7.28.3.7 setDescription()

```
void es.deusto.server.data.Post.setDescription (
    String description )
```

### 7.28.3.8 `setEventName()`

```
void es.deusto.server.data.Post.setEventName (
    String eventName )
```

### 7.28.3.9 `setOrganizerEmail()`

```
void es.deusto.server.data.Post.setOrganizerEmail (
    String organizerEmail )
```

### 7.28.3.10 `setTitle()`

```
void es.deusto.server.data.Post.setTitle (
    String title )
```

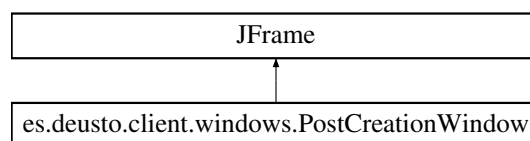
The documentation for this class was generated from the following file:

- [src/main/java/es/deusto/server/data/Post.java](#)

## 7.29 `es.deusto.client.windows.PostCreationWindow` Class Reference

This class is the one used to create a window that allows an organizer to have like a double check of the event he has just created.

Inheritance diagram for `es.deusto.client.windows.PostCreationWindow`:



### Public Member Functions

- [PostCreationWindow](#) ([Controller](#) controller, [EventInfo](#) event)

### 7.29.1 Detailed Description

This class is the one used to create a window that allows an organizer to have like a double check of the event he has just created.

## 7.29.2 Constructor & Destructor Documentation

### 7.29.2.1 PostCreationWindow()

```
es.deusto.client.windows.PostCreationWindow.PostCreationWindow (
    Controller controller,
    EventInfo event )
```

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

- src/main/java/es/deusto/client/windows/[PostCreationWindow.java](#)

## 7.30 es.deusto.server.dao.PostDAO Class Reference

This classes is the one used to create a DAO for a post.

### Public Member Functions

- ArrayList< [Post](#) > [getPostsByEvent](#) ([Event](#) event)
- void [storePost](#) ([Post](#) post)
- void [updatePost](#) ([Post](#) post)
- void [deletePost](#) ([Post](#) post)

### Protected Member Functions

- [PostDAO](#) (PersistenceManager pm)
- PersistenceManager [getPersistenceManager](#) ()
- void [close](#) ()

### 7.30.1 Detailed Description

This classes is the one used to create a DAO for a post.

## 7.30.2 Constructor & Destructor Documentation

### 7.30.2.1 PostDAO()

```
es.deusto.server.dao.PostDAO.PostDAO (
    PersistenceManager pm ) [protected]
```

## 7.30.3 Member Function Documentation

### 7.30.3.1 close()

```
void es.deusto.server.dao.PostDAO.close ( ) [protected]
```

### 7.30.3.2 deletePost()

```
void es.deusto.server.dao.PostDAO.deletePost (
    Post post )
```

### 7.30.3.3 getPersistenceManager()

```
PersistenceManager es.deusto.server.dao.PostDAO.getPersistenceManager ( ) [protected]
```

### 7.30.3.4 getPostsByEvent()

```
ArrayList<Post> es.deusto.server.dao.PostDAO.getPostsByEvent (
    Event event )
```

### 7.30.3.5 storePost()

```
void es.deusto.server.dao.PostDAO.storePost (
    Post post )
```



### 7.30.3.6 updatePost()

```
void es.deusto.server.dao.PostDAO.updatePost (
    Post post )
```

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

- [src/main/java/es/deusto/server/dao/PostDAO.java](#)

## 7.31 es.deusto.serialization.PostInfo Class Reference

DTO equivalent class of the Post class from the server.

### Public Member Functions

- [PostInfo](#) ()
- [PostInfo](#) ([Post](#) post)
- String [getTitle](#) ()
- void [setTitle](#) (String title)
- String [getDescription](#) ()
- void [setDescription](#) (String description)
- Date [getDate](#) ()
- void [setDate](#) (Date date)
- String [getEventName](#) ()
- void [setEventName](#) (String eventName)
- String [getOrganizerEmail](#) ()
- void [setOrganizerEmail](#) (String organizerEmail)

### 7.31.1 Detailed Description

DTO equivalent class of the Post class from the server.

This class is used to transport all the relevant information of a post through the network.

Since

Sprint 2

### 7.31.2 Constructor & Destructor Documentation

#### 7.31.2.1 PostInfo() [1/2]

```
es.deusto.serialization.PostInfo.PostInfo ( )
```

### 7.31.2.2 PostInfo() [2/2]

```
es.deusto.serialization.PostInfo.PostInfo (
    Post post )
```

## 7.31.3 Member Function Documentation

### 7.31.3.1 getDate()

```
Date es.deusto.serialization.PostInfo.getDate ( )
```

### 7.31.3.2 getDescription()

```
String es.deusto.serialization.PostInfo.getDescription ( )
```

### 7.31.3.3 getEventName()

```
String es.deusto.serialization.PostInfo.getEventName ( )
```

### 7.31.3.4 getOrganizerEmail()

```
String es.deusto.serialization.PostInfo.getOrganizerEmail ( )
```

### 7.31.3.5 getTitle()

```
String es.deusto.serialization.PostInfo.getTitle ( )
```

### 7.31.3.6 setDate()

```
void es.deusto.serialization.PostInfo.setDate (
    Date date )
```

### 7.31.3.7 setDescription()

```
void es.deusto.serialization.PostInfo.setDescription (
    String description )
```

### 7.31.3.8 setEventName()

```
void es.deusto.serialization.PostInfo.setEventName (
    String eventName )
```

### 7.31.3.9 setOrganizerEmail()

```
void es.deusto.serialization.PostInfo.setOrganizerEmail (
    String organizerEmail )
```

### 7.31.3.10 setTitle()

```
void es.deusto.serialization.PostInfo.setTitle (
    String title )
```

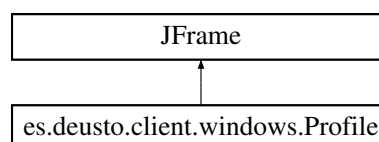
The documentation for this class was generated from the following file:

- [src/main/java/es/deusto/serialization/PostInfo.java](#)

## 7.32 es.deusto.client.windows.Profile Class Reference

This class is the one used to create a window that allows a user/organizer to have a profile with all of his information.

Inheritance diagram for es.deusto.client.windows.Profile:



### Public Member Functions

- [Profile](#) ([Controller](#) controller, JFrame mainWindow)

### 7.32.1 Detailed Description

This class is the one used to create a window that allows a user/organizer to have a profile with all of his information.

### 7.32.2 Constructor & Destructor Documentation

#### 7.32.2.1 Profile()

```
es.deusto.client.windows.Profile.Profile (
    Controller controller,
    JFrame mainWindow )
```

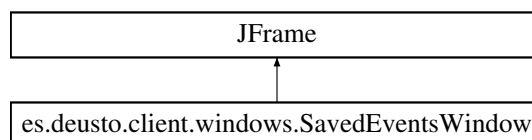
The documentation for this class was generated from the following file:

- [src/main/java/es/deusto/client/windows/Profile.java](#)

## 7.33 es.deusto.client.windows.SavedEventsWindow Class Reference

This class is the one used to create a window that allows to scroll, search and choose from all of the events stored in the database.

Inheritance diagram for es.deusto.client.windows.SavedEventsWindow:



### Public Member Functions

- [SavedEventsWindow](#) ([Controller](#) controller)

### 7.33.1 Detailed Description

This class is the one used to create a window that allows to scroll, search and choose from all of the events stored in the database.

### 7.33.2 Constructor & Destructor Documentation

### 7.33.2.1 SavedEventsWindow()

```
es.deusto.client.windows.SavedEventsWindow.SavedEventsWindow (
    Controller controller )
```

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

- src/main/java/es/deusto/client/windows/[SavedEventsWindow.java](#)

## 7.34 es.deusto.junit.serialization.SerializationTest Class Reference

This classes is the one that tests funcionalities such as creating an event, creating a user, orginzer, crea ting posts and topics.

### Public Member Functions

- void [beforeClass](#) ()
- void [testEventInfoFromEvent](#) ()
- void [testEventInfoSetters](#) ()
- void [testOrganizerInfoFromOrganizer](#) ()
- void [testOrganizerInfoSetters](#) ()
- void [testPostInfoFromPost](#) ()
- void [testPostInfoSetters](#) ()
- void [testUserInfoFromUser](#) ()
- void [testUserInfoSetters](#) ()
- void [testUserInfoStringConstructor](#) ()
- void [testSignupAttemptUser](#) ()
- void [testSignupAttemptOrganizer](#) ()

### Static Public Member Functions

- static void [startUp](#) ()
- static junit.framework.Test [suite](#) ()

### 7.34.1 Detailed Description

This classes is the one that tests funcionalities such as creating an event, creating a user, orginzer, crea ting posts and topics.

Is the class that checks that all the funcionalities work as intended.

### 7.34.2 Member Function Documentation

#### 7.34.2.1 beforeClass()

```
void es.deusto.junit.serialization.SerializationTest.beforeClass ( )
```

#### 7.34.2.2 startUp()

```
static void es.deusto.junit.serialization.SerializationTest.startUp ( ) [static]
```

#### 7.34.2.3 suite()

```
static junit.framework.Test es.deusto.junit.serialization.SerializationTest.suite ( ) [static]
```

#### 7.34.2.4 testEventInfoFromEvent()

```
void es.deusto.junit.serialization.SerializationTest.testEventInfoFromEvent ( )
```

#### 7.34.2.5 testEventInfoSetters()

```
void es.deusto.junit.serialization.SerializationTest.testEventInfoSetters ( )
```

#### 7.34.2.6 testOrganizerInfoFromOrganizer()

```
void es.deusto.junit.serialization.SerializationTest.testOrganizerInfoFromOrganizer ( )
```

#### 7.34.2.7 testOrganizerInfoSetters()

```
void es.deusto.junit.serialization.SerializationTest.testOrganizerInfoSetters ( )
```

#### 7.34.2.8 testPostInfoFromPost()

```
void es.deusto.junit.serialization.SerializationTest.testPostInfoFromPost ( )
```

#### 7.34.2.9 testPostInfoSetters()

```
void es.deusto.junit.serialization.SerializationTest.testPostInfoSetters ( )
```

#### 7.34.2.10 testSignupAttemptOrganizer()

```
void es.deusto.junit.serialization.SerializationTest.testSignupAttemptOrganizer ( )
```

#### 7.34.2.11 testSignupAttemptUser()

```
void es.deusto.junit.serialization.SerializationTest.testSignupAttemptUser ( )
```

#### 7.34.2.12 testUserInfoFromUser()

```
void es.deusto.junit.serialization.SerializationTest.testUserInfoFromUser ( )
```

#### 7.34.2.13 testUserInfoSetters()

```
void es.deusto.junit.serialization.SerializationTest.testUserInfoSetters ( )
```

#### 7.34.2.14 testUserInfoStringConstructor()

```
void es.deusto.junit.serialization.SerializationTest.testUserInfoStringConstructor ( )
```

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

- [src/test/java/es/deusto/junit/serialization/SerializationTest.java](#)

## 7.35 es.deusto.server.server.Server Class Reference

This class is the one that receives the requests from the client.

## Public Member Functions

- [Server](#) ()
- Response [attemptNormalLogin](#) ([LoginAttempt](#) login)  
*This is used for the login process of regular users.*
- Response [attemptOrganizerLogin](#) ([LoginAttempt](#) login)  
*This method is used for the login process of organizers.*
- Response [attemptNormalSignup](#) ([SignupAttempt](#) signup)  
*This method is invoked whenever a POST request is made to the following path: /signup .*
- Response [attemptOrganizerSignup](#) ([SignupAttempt](#) signup)  
*This method is used of the signup method of Organizers.*
- Response [attemptNormalUpdate](#) ([SignupAttempt](#) signup)  
*This method is used to update the information of regular users' account.*
- Response [attemptOrganizerUpdate](#) ([SignupAttempt](#) signup)  
*This method is used to update the information of an Organizer account.*
- Response [getRecommendedEvents](#) ([SignupAttempt](#) signupAttempt)  
*This method is used for recommend Event to users.*
- Response [recoverPassword](#) ([LoginAttempt](#) userInfo)  
*This method is used for password recovery for users who can't remember their passwords.*
- Response [deleteEvent](#) ([EventInfo](#) eventInfo)  
*This method is used for the removal of events.*
- Response [createEvent](#) ([EventInfo](#) eventInfo)  
*This method is used for the creation of new events from a given organizer.*
- Response [createPost](#) ([PostInfo](#) postInfo)  
*This method is used for the creation of new Post from a given Event.*
- Response [attemptUserDelete](#) ([UserInfo](#) userInfo)  
*This method is used of the delete method of regular Users.*
- Response [attemptOrganizerDelete](#) ([OrganizerInfo](#) organizerInfo)  
*This method is used of the delete method of Organizers.*
- Response [sayHello](#) ()  
*method used for manual testing*

### 7.35.1 Detailed Description

This class is the one that receives the requests from the client.

There is no functionality implemented in this class, all the possible requests have a corresponding method inside the [AppService](#) class in which all the functionality is implemented. That way, the [Server](#) class is abstracted from the business objects.

### 7.35.2 Constructor & Destructor Documentation

#### 7.35.2.1 Server()

```
es.deusto.server.server.Server.Server ( )
```



### 7.35.3 Member Function Documentation

#### 7.35.3.1 attemptNormalLogin()

```
Response es.deusto.server.server.Server.attemptNormalLogin (
    LoginAttempt login )
```

This is used for the login process of regular users.

It is invoked whenever a POST request is made to the following path: /login. It is used for the login process of regular users.

##### Parameters

<i>login</i>	LoginAttempt class containing the information of the user needed for its log in (email and password).
--------------	---

##### Since

Sprint 1.

##### Returns

Response object with the information of the user, in case of valid credentials. If not, an empty response is returned.

#### 7.35.3.2 attemptNormalSignup()

```
Response es.deusto.server.server.Server.attemptNormalSignup (
    SignupAttempt signup )
```

This method is invoked whenever a POST request is made to the following path: /signup .

It is used of the signup method of regular users.

##### Parameters

<i>signup</i>	Signup information of the user (name, email, password, city and interests) contained in a SignupAttempt object.
---------------	---

##### Returns

Response object with the information of the user, in case of correct user registration. If not, an empty response is returned

Since

Sprint 1

### 7.35.3.3 attemptNormalUpdate()

```
Response es.deusto.server.server.Server.attemptNormalUpdate (
    SignupAttempt signup )
```

This method is used to update the information of regular users' account.

It is invoked whenever a POST request is made to the following path: /update .

#### Parameters

<i>signup</i>	SignupAttempt object containing the information of the User (name, email, password, city and interests). In this case it is also used for updating their information since its basic structure doesn't change.
---------------	--

Since

Sprint 2

#### Returns

Response object with the information of the user, in case of correct user registration. If not, an empty response is returned

### 7.35.3.4 attemptOrganizerDelete()

```
Response es.deusto.server.server.Server.attemptOrganizerDelete (
    OrganizerInfo organizerInfo )
```

This method is used of the delete method of Organizers.

It is invoked whenever a POST request is made to the following path: /delete .

#### Parameters

<i>organizerInfo</i>	OrganizerInfo object containing Organizer's email.
----------------------	--

Since

Sprint 2

**Returns**

Response string with the Organizer's email, in case the delete was successful. If not, an empty response is returned.

**7.35.3.5 attemptOrganizerLogin()**

```
Response es.deusto.server.server.Server.attemptOrganizerLogin (
    LoginAttempt login )
```

This method is used for the login process of organizers.

It is invoked whenever a POST request is made to the following path: /loginOrganizer.

**Parameters**

<i>login</i>	LoginAttempt class containing the information of the organizer needed for its log in (email and password).
--------------	--

**Returns**

Response object with the information of the organizer, in case of valid credentials. If not, an empty response is returned.

**Since**

Sprint 2

**7.35.3.6 attemptOrganizerSignup()**

```
Response es.deusto.server.server.Server.attemptOrganizerSignup (
    SignupAttempt signup )
```

This method is used of the signup method of Organizers.

It is invoked whenever a POST request is made to the following path: /signupOrganizer.

**Parameters**

<i>signup</i>	SignupAttempt object containing the information of the organizer requested when registering for the first time (name, email, password and organization).
---------------	--

**Returns**

Response object with the information of the organizer, in case of correct registration. If not, an empty response is returned.

Since

Sprint 2

### 7.35.3.7 attemptOrganizerUpdate()

```
Response es.deusto.server.server.Server.attemptOrganizerUpdate (
    SignupAttempt signup )
```

This method is used to update the information of an Organizer account.

It is invoked whenever a POST request is made to the following path: /updateOrganizer .

#### Parameters

<i>signup</i>	SignupAttempt object containing the information of the Organizers (name, email, password and organization). In this case it is also used for updating their information since its basic structure doesn't change.
---------------	---

Since

Sprint 2

#### Returns

Response object with the information of the organizer, in case of correct registration. If not, an empty response is returned

### 7.35.3.8 attemptUserDelete()

```
Response es.deusto.server.server.Server.attemptUserDelete (
    UserInfo userInfo )
```

This method is used of the delete method of regular Users.

It is invoked whenever a POST request is made to the following path: /delete .

#### Parameters

<i>userInfo</i>	UserInfo object containing user's email.
-----------------	--

Since

Sprint 2

**Returns**

Response string with the user's email, in case of user delete was successful. If not, an empty response is returned.

**7.35.3.9 createEvent()**

```
Response es.deusto.server.server.Server.createEvent (
    EventInfo eventInfo )
```

This method is used for the creation of new events from a given organizer.

It is invoked whenever a POST request is made to the following path: /createEvent.

**Parameters**

<i>eventInfo</i>	EventInfo object with the data of the event that will be created.
------------------	---

**Returns**

Response object containing the information of a EventInfo.

**Since**

Sprint 2

**7.35.3.10 createPost()**

```
Response es.deusto.server.server.Server.createPost (
    PostInfo postInfo )
```

This method is used for the creation of new Post from a given Event.

It is invoked whenever a POST request is made to the following path: /createEvent.

**Parameters**

<i>postInfo</i>	PostInfo object with the data of the event that will be created.
-----------------	--

**Returns**

Response object containing the information of a PostInfo.

Since

Sprint 2

#### 7.35.3.11 deleteEvent()

```
Response es.deusto.server.server.Server.deleteEvent (
    EventInfo eventInfo )
```

This method is used for the removal of events.

It is invoked whenever a POST request is made to the following path: /deleteEvent .

Parameters

<i>eventInfo</i>	information of the event to be deleted
------------------	--

Returns

Response object

#### 7.35.3.12 getRecommendedEvents()

```
Response es.deusto.server.server.Server.getRecommendedEvents (
    SignupAttempt signupAttempt )
```

This method is used for recommend Event to users.

It is invoked whenever a POST request is made to the following path: /recomendation .

Parameters

<i>signupAttempt</i>	SignupAttempt object containing the interests and locations of the User.
----------------------	--

Returns

Response object with the events information

Since

Sprint 3

### 7.35.3.13 recoverPassword()

```
Response es.deusto.server.server.Server.recoverPassword (
    LoginAttempt userInfo )
```

This method is used for password recovery for users who can't remember their passwords.

It is invoked whenever a POST request is made to the following path: /passwordRecovery .

#### Parameters

<i>userInfo</i>	LoginAttempt object with the email of the user who needs a new password.
-----------------	--

#### Returns

Response object with the following message: "OK".

#### Since

Sprint 1

### 7.35.3.14 sayHello()

```
Response es.deusto.server.server.Server.sayHello ( )
```

method used for manual testing

#### Returns

#### Deprecated

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

- src/main/java/es/deusto/server/server/[Server.java](#)

## 7.36 es.deusto.junit.server.ServerTest Class Reference

This classes is used to create several tests to see if the server is working as intended.

## Public Member Functions

- void [testNormalSignup](#) ()
- void [testNormalUpdate](#) ()
- void [testOrganizerSignup](#) ()
- void [testOrganizerLogin](#) ()
- void [testOrganizerUpdate](#) ()
- void [testCreateEvent](#) ()
- void [testCreatePost](#) ()

## Static Public Member Functions

- static void [startUp](#) ()
- static junit.framework.Test [suite](#) ()
- static void [setUp](#) ()
- static void [tearDown](#) ()

### 7.36.1 Detailed Description

This classes is used to create several tests to see if the server is working as intended.

### 7.36.2 Member Function Documentation

#### 7.36.2.1 [setUp\(\)](#)

```
static void es.deusto.junit.server.ServerTest.setUp ( ) [static]
```

#### 7.36.2.2 [startUp\(\)](#)

```
static void es.deusto.junit.server.ServerTest.startUp ( ) [static]
```

#### 7.36.2.3 [suite\(\)](#)

```
static junit.framework.Test es.deusto.junit.server.ServerTest.suite ( ) [static]
```



#### 7.36.2.4 tearDown()

```
static void es.deusto.junit.server.ServerTest.tearDown ( ) [static]
```

#### 7.36.2.5 testCreateEvent()

```
void es.deusto.junit.server.ServerTest.testCreateEvent ( )
```

#### 7.36.2.6 testCreatePost()

```
void es.deusto.junit.server.ServerTest.testCreatePost ( )
```

#### 7.36.2.7 testNormalSignup()

```
void es.deusto.junit.server.ServerTest.testNormalSignup ( )
```

#### 7.36.2.8 testNormalUpdate()

```
void es.deusto.junit.server.ServerTest.testNormalUpdate ( )
```

#### 7.36.2.9 testOrganizerLogin()

```
void es.deusto.junit.server.ServerTest.testOrganizerLogin ( )
```

#### 7.36.2.10 testOrganizerSignup()

```
void es.deusto.junit.server.ServerTest.testOrganizerSignup ( )
```

### 7.36.2.11 testOrganizerUpdate()

```
void es.deusto.junit.server.ServerTest.testOrganizerUpdate ( )
```

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

- [src/test/java/es/deusto/junit/server/ServerTest.java](#)

## 7.37 es.deusto.serialization.SignupAttempt Class Reference

This class encapsulates all the information required when a user (Either normal User or Organizer) signs up.

### Public Member Functions

- [SignupAttempt](#) ()
- [SignupAttempt](#) ([UserInfo](#) userInfo)
- String [getEmail](#) ()
- void [setEmail](#) (String email)
- String [getPassword](#) ()
- void [setPassword](#) (String password)
- String [getName](#) ()
- void [setName](#) (String name)
- String [getCity](#) ()
- void [setCity](#) (String city)
- String [getOrganization](#) ()
- void [setOrganization](#) (String organization)
- ArrayList< [TopicInfo](#) > [getInterests](#) ()
- void [setInterests](#) (ArrayList< [TopicInfo](#) > interests)
- ArrayList< [EventInfo](#) > [getSavedEvents](#) ()
- void [setSaveEvents](#) (ArrayList< [EventInfo](#) > savedEvents)
- [User](#) [buildUser](#) ()  
*Builds an instance of the business object User.*
- [Organizer](#) [buildOrganizer](#) ()  
*Builds an instance of the business object Organizer.*
- String [getCountry](#) ()
- void [setCountry](#) (String country)
- String [toString](#) ()

### 7.37.1 Detailed Description

This class encapsulates all the information required when a user (Either normal User or Organizer) signs up.

Since

Sprint 1

### 7.37.2 Constructor & Destructor Documentation

### 7.37.2.1 SignupAttempt() [1/2]

```
es.deusto.serialization.SignupAttempt.SignupAttempt ( )
```

### 7.37.2.2 SignupAttempt() [2/2]

```
es.deusto.serialization.SignupAttempt.SignupAttempt (
    UserInfo userInfo )
```

## 7.37.3 Member Function Documentation

### 7.37.3.1 buildOrganizer()

```
Organizer es.deusto.serialization.SignupAttempt.buildOrganizer ( )
```

Builds an instance of the business object Organizer.

The main purpose of this function is prevent object repetition. It's only used in the server.

#### Returns

instance of Organizer.

### 7.37.3.2 buildUser()

```
User es.deusto.serialization.SignupAttempt.buildUser ( )
```

Builds an instance of the business object User.

The main purpose of this function is prevent object repetition. It's only used in the server.

#### Returns

instance of es.deusto.server.data.User.java

### 7.37.3.3 getCity()

```
String es.deusto.serialization.SignupAttempt.getCity ( )
```

#### 7.37.3.4 `getCountry()`

```
String es.deusto.serialization.SignupAttempt.getCountry ( )
```

#### 7.37.3.5 `getEmail()`

```
String es.deusto.serialization.SignupAttempt.getEmail ( )
```

#### 7.37.3.6 `getInterests()`

```
ArrayList<TopicInfo> es.deusto.serialization.SignupAttempt.getInterests ( )
```

#### 7.37.3.7 `getName()`

```
String es.deusto.serialization.SignupAttempt.getName ( )
```

#### 7.37.3.8 `getOrganization()`

```
String es.deusto.serialization.SignupAttempt.getOrganization ( )
```

#### 7.37.3.9 `getPassword()`

```
String es.deusto.serialization.SignupAttempt.getPassword ( )
```

#### 7.37.3.10 `getSavedEvents()`

```
ArrayList<EventInfo> es.deusto.serialization.SignupAttempt.getSavedEvents ( )
```

#### 7.37.3.11 `setCity()`

```
void es.deusto.serialization.SignupAttempt.setCity (
    String city )
```

**7.37.3.12 setCountry()**

```
void es.deusto.serialization.SignupAttempt.setCountry (
    String country )
```

**7.37.3.13 setEmail()**

```
void es.deusto.serialization.SignupAttempt.setEmail (
    String email )
```

**7.37.3.14 setInterests()**

```
void es.deusto.serialization.SignupAttempt.setInterests (
    ArrayList< TopicInfo > interests )
```

**7.37.3.15 setName()**

```
void es.deusto.serialization.SignupAttempt.setName (
    String name )
```

**7.37.3.16 setOrganization()**

```
void es.deusto.serialization.SignupAttempt.setOrganization (
    String organization )
```

**7.37.3.17 setPassword()**

```
void es.deusto.serialization.SignupAttempt.setPassword (
    String password )
```

**7.37.3.18 setSaveEvents()**

```
void es.deusto.serialization.SignupAttempt.setSaveEvents (
    ArrayList< EventInfo > savedEvents )
```

### 7.37.3.19 toString()

```
String es.deusto.serialization.SignupAttempt.toString ( )
```

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

- [src/main/java/es/deusto/serialization/SignupAttempt.java](#)

## 7.38 es.deusto.server.data.Topic Class Reference

This classes is the one that defines a topic and it is used to create one.

### Public Member Functions

- [Topic](#) ()
- [Topic](#) ([TopicInfo](#) info)
- String [getName](#) ()
- void [setName](#) (String name)
- String [toString](#) ()

### 7.38.1 Detailed Description

This classes is the one that defines a topic and it is used to create one.

### 7.38.2 Constructor & Destructor Documentation

#### 7.38.2.1 Topic() [1/2]

```
es.deusto.server.data.Topic.Topic ( )
```

#### 7.38.2.2 Topic() [2/2]

```
es.deusto.server.data.Topic.Topic (
    TopicInfo info )
```

### 7.38.3 Member Function Documentation

### 7.38.3.1 getName()

```
String es.deusto.server.data.Topic.getName ( )
```

### 7.38.3.2 setName()

```
void es.deusto.server.data.Topic.setName (
    String name )
```

### 7.38.3.3 toString()

```
String es.deusto.server.data.Topic.toString ( )
```

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

- [src/main/java/es/deusto/server/data/Topic.java](#)

## 7.39 es.deusto.server.dao.TopicDAO Class Reference

This classes is the one used to create a DAO a topic.

### Public Member Functions

- [Topic](#) [getTopic](#) (String name)
- void [storeTopic](#) ([Topic](#) topic)
- void [updateTopic](#) ([Topic](#) topic)
- void [deleteTopic](#) (String name)

### Protected Member Functions

- [TopicDAO](#) (PersistenceManager pm)
- PersistenceManager [getPersistenceManager](#) ()
- void [close](#) ()

### 7.39.1 Detailed Description

This classes is the one used to create a DAO a topic.

## 7.39.2 Constructor & Destructor Documentation

### 7.39.2.1 TopicDAO()

```
es.deusto.server.dao.TopicDAO.TopicDAO (
    PersistenceManager pm ) [protected]
```

## 7.39.3 Member Function Documentation

### 7.39.3.1 close()

```
void es.deusto.server.dao.TopicDAO.close ( ) [protected]
```

### 7.39.3.2 deleteTopic()

```
void es.deusto.server.dao.TopicDAO.deleteTopic (
    String name )
```

### 7.39.3.3 getPersistenceManager()

```
PersistenceManager es.deusto.server.dao.TopicDAO.getPersistenceManager ( ) [protected]
```

### 7.39.3.4 getTopic()

```
Topic es.deusto.server.dao.TopicDAO.getTopic (
    String name )
```

### 7.39.3.5 storeTopic()

```
void es.deusto.server.dao.TopicDAO.storeTopic (
    Topic topic )
```



### 7.39.3.6 updateTopic()

```
void es.deusto.server.dao.TopicDAO.updateTopic (
    Topic topic )
```

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

- src/main/java/es/deusto/server/dao/TopicDAO.java

## 7.40 es.deusto.serialization.TopicInfo Class Reference

DTO equivalent class of the Topic class from the server.

### Public Member Functions

- TopicInfo ()
- TopicInfo (String name)
- TopicInfo (Topic c)
- String getName ()
- void setName (String name)
- String toString ()

### Static Public Member Functions

- static Topic parseTopic (TopicInfo topicInfo)  
*Creates a Topic object from a TopicInfo one.*

### 7.40.1 Detailed Description

DTO equivalent class of the Topic class from the server.

This class is used to transport all the relevant information of a topic through the network.

Since

Sprint 1

### 7.40.2 Constructor & Destructor Documentation

#### 7.40.2.1 TopicInfo() [1/3]

```
es.deusto.serialization.TopicInfo.TopicInfo ( )
```

#### 7.40.2.2 TopicInfo() [2/3]

```
es.deusto.serialization.TopicInfo.TopicInfo (
    String name )
```

#### 7.40.2.3 TopicInfo() [3/3]

```
es.deusto.serialization.TopicInfo.TopicInfo (
    Topic c )
```

### 7.40.3 Member Function Documentation

#### 7.40.3.1 getName()

```
String es.deusto.serialization.TopicInfo.getName ( )
```

#### 7.40.3.2 parseTopic()

```
static Topic es.deusto.serialization.TopicInfo.parseTopic (
    TopicInfo topicInfo ) [static]
```

Creates a Topic object from a [TopicInfo](#) one.

##### Parameters

<i>topicInfo</i>	the source of the information
------------------	-------------------------------

##### Returns

an Object Topic

#### 7.40.3.3 setName()

```
void es.deusto.serialization.TopicInfo.setName (
    String name )
```

#### 7.40.3.4 toString()

```
String es.deusto.serialization.TopicInfo.toString ( )
```

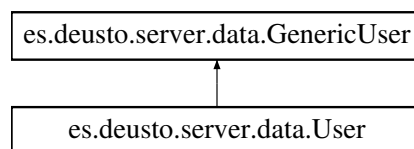
The documentation for this class was generated from the following file:

- [src/main/java/es/deusto/serialization/TopicInfo.java](#)

## 7.41 es.deusto.server.data.User Class Reference

This classes is the one that defines a normal user and it is used to create one.

Inheritance diagram for es.deusto.server.data.User:



### Public Member Functions

- String [getCity](#) ()
- String [getCountry](#) ()
- void [setCountry](#) (String country)
- void [setCity](#) (String city)
- ArrayList< [Event](#) > [getSavedEvents](#) ()
- void [setSavedEvents](#) (ArrayList< [Event](#) > savedEvents)
- void [addEvent](#) ([Event](#) event)
- ArrayList< [Topic](#) > [getInterests](#) ()
- void [setInterests](#) (ArrayList< [Topic](#) > topics)
- void [addInterest](#) ([Topic](#) topic)
- String [toString](#) ()

#### 7.41.1 Detailed Description

This classes is the one that defines a normal user and it is used to create one.

#### 7.41.2 Member Function Documentation

#### 7.41.2.1 addEvent()

```
void es.deusto.server.data.User.addEvent (
    Event event )
```

#### 7.41.2.2 addInterest()

```
void es.deusto.server.data.User.addInterest (
    Topic topic )
```

#### 7.41.2.3 getCity()

```
String es.deusto.server.data.User.getCity ( )
```

#### 7.41.2.4 getCountry()

```
String es.deusto.server.data.User.getCountry ( )
```

#### 7.41.2.5 getInterests()

```
ArrayList<Topic> es.deusto.server.data.User.getInterests ( )
```

#### 7.41.2.6 getSavedEvents()

```
ArrayList<Event> es.deusto.server.data.User.getSavedEvents ( )
```

#### 7.41.2.7 setCity()

```
void es.deusto.server.data.User.setCity (
    String city )
```

### 7.41.2.8 setCountry()

```
void es.deusto.server.data.User.setCountry (
    String country )
```

### 7.41.2.9 setInterests()

```
void es.deusto.server.data.User.setInterests (
    ArrayList< Topic > topics )
```

### 7.41.2.10 setSavedEvents()

```
void es.deusto.server.data.User.setSavedEvents (
    ArrayList< Event > savedEvents )
```

### 7.41.2.11 toString()

```
String es.deusto.server.data.User.toString ( )
```

Reimplemented from [es.deusto.server.data.GenericUser](#).

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

- [src/main/java/es/deusto/server/data/User.java](#)

## 7.42 es.deusto.server.dao.UserDAO Class Reference

This classes is the one used to create an user DAO.

### Public Member Functions

- ArrayList< [User](#) > [getAllUsers](#) ()
- [User](#) [getUser](#) (String email)
- void [storeUser](#) ([User](#) user)
- void [updateUser](#) ([User](#) user)
- void [deleteUser](#) (String email)

### Protected Member Functions

- [UserDAO](#) (PersistenceManager pm)
- PersistenceManager [getPersistenceManager](#) ()
- void [close](#) ()

### 7.42.1 Detailed Description

This classes is the one used to create an user DAO.

### 7.42.2 Constructor & Destructor Documentation

#### 7.42.2.1 UserDao()

```
es.deusto.server.dao.UserDAO.UserDAO (
    PersistenceManager pm ) [protected]
```

### 7.42.3 Member Function Documentation

#### 7.42.3.1 close()

```
void es.deusto.server.dao.UserDAO.close ( ) [protected]
```

#### 7.42.3.2 deleteUser()

```
void es.deusto.server.dao.UserDAO.deleteUser (
    String email )
```

#### 7.42.3.3 getAllUsers()

```
ArrayList<User> es.deusto.server.dao.UserDAO.getAllUsers ( )
```

#### 7.42.3.4 getPersistenceManager()

```
PersistenceManager es.deusto.server.dao.UserDAO.getPersistenceManager ( ) [protected]
```

### 7.42.3.5 getUser()

```
User es.deusto.server.dao.UserDAO.getUser (
    String email )
```

### 7.42.3.6 storeUser()

```
void es.deusto.server.dao.UserDAO.storeUser (
    User user )
```

### 7.42.3.7 updateUser()

```
void es.deusto.server.dao.UserDAO.updateUser (
    User user )
```

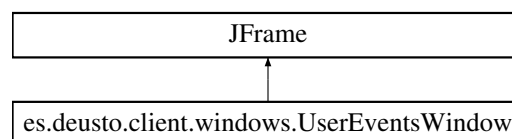
The documentation for this class was generated from the following file:

- [src/main/java/es/deusto/server/dao/UserDAO.java](#)

## 7.43 es.deusto.client.windows.UserEventsWindow Class Reference

This class is the one used to create a window that allows a user to scroll, search and select from a list of events he previously has chosen from.

Inheritance diagram for es.deusto.client.windows.UserEventsWindow:



### Public Member Functions

- [UserEventsWindow](#) ([Controller](#) controller)

### 7.43.1 Detailed Description

This class is the one used to create a window that allows a user to scroll, search and select from a list of events he previously has chosen from.

A personal event list.

## 7.43.2 Constructor & Destructor Documentation

### 7.43.2.1 UserEventsWindow()

```
es.deusto.client.windows.UserEventsWindow.UserEventsWindow (
    Controller controller )
```

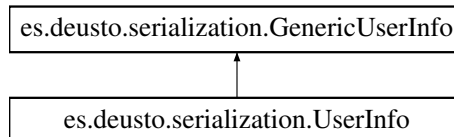
The documentation for this class was generated from the following file:

- [src/main/java/es/deusto/client/windows/UserEventsWindow.java](#)

## 7.44 es.deusto.serialization.UserInfo Class Reference

DTO equivalent class of the User class from the server.

Inheritance diagram for es.deusto.serialization.UserInfo:



### Public Member Functions

- [UserInfo](#) (String name, String email, String city, String country)
- [UserInfo](#) ()
- [UserInfo](#) ([User](#) user)
- String [getName](#) ()
- void [setName](#) (String name)
- String [getEmail](#) ()
- void [setEmail](#) (String email)
- String [getCity](#) ()
- void [setCity](#) (String city)
- ArrayList< [EventInfo](#) > [getSavedEvents](#) ()
- void [setSavedEvents](#) (ArrayList< [EventInfo](#) > savedEvents)
- ArrayList< [TopicInfo](#) > [getInterests](#) ()
- void [setInterests](#) (ArrayList< [TopicInfo](#) > interests)
- void [addInterest](#) ([TopicInfo](#) interest)
- String [getCountry](#) ()
- void [setCountry](#) (String country)



### 7.44.1 Detailed Description

DTO equivalent class of the User class from the server.

This class is used to transport all the relevant information of a user through the network.

Since

Sprint 1

### 7.44.2 Constructor & Destructor Documentation

#### 7.44.2.1 UserInfo() [1/3]

```
es.deusto.serialization.UserInfo.UserInfo (
    String name,
    String email,
    String city,
    String country )
```

#### 7.44.2.2 UserInfo() [2/3]

```
es.deusto.serialization.UserInfo.UserInfo ( )
```

#### 7.44.2.3 UserInfo() [3/3]

```
es.deusto.serialization.UserInfo.UserInfo (
    User user )
```

### 7.44.3 Member Function Documentation

#### 7.44.3.1 addInterest()

```
void es.deusto.serialization.UserInfo.addInterest (
    TopicInfo interest )
```

**7.44.3.2   getCity()**

```
String es.deusto.serialization.UserInfo.getCity ( )
```

**7.44.3.3   getCountry()**

```
String es.deusto.serialization.UserInfo.getCountry ( )
```

**7.44.3.4   getEmail()**

```
String es.deusto.serialization.UserInfo.getEmail ( )
```

**7.44.3.5   getInterests()**

```
ArrayList<TopicInfo> es.deusto.serialization.UserInfo.getInterests ( )
```

**7.44.3.6   getName()**

```
String es.deusto.serialization.UserInfo.getName ( )
```

**7.44.3.7   getSavedEvents()**

```
ArrayList<EventInfo> es.deusto.serialization.UserInfo.getSavedEvents ( )
```

**7.44.3.8   setCity()**

```
void es.deusto.serialization.UserInfo.setCity (
    String city )
```

#### 7.44.3.9 setCountry()

```
void es.deusto.serialization.UserInfo.setCountry (
    String country )
```

#### 7.44.3.10 setEmail()

```
void es.deusto.serialization.UserInfo.setEmail (
    String email )
```

#### 7.44.3.11 setInterests()

```
void es.deusto.serialization.UserInfo.setInterests (
    ArrayList< TopicInfo > interests )
```

#### 7.44.3.12 setName()

```
void es.deusto.serialization.UserInfo.setName (
    String name )
```

#### 7.44.3.13 setSavedEvents()

```
void es.deusto.serialization.UserInfo.setSavedEvents (
    ArrayList< EventInfo > savedEvents )
```

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

- [src/main/java/es/deusto/serialization/UserInfo.java](#)



## Chapter 8

# File Documentation

### 8.1 src/main/java/es/deusto/client/controller/Controller.java File Reference

#### Classes

- class [es.deusto.client.controller.Controller](#)  
*This class is the one carrying out all the communication with the server.*

#### Packages

- package [es.deusto.client.controller](#)

### 8.2 src/main/java/es/deusto/client/main/Main.java File Reference

#### Classes

- class [es.deusto.client.main.Main](#)  
*This class is the one that starts the application.*

#### Packages

- package [es.deusto.client.main](#)

### 8.3 src/main/java/es/deusto/client/windows/ChangePass.java File Reference

#### Classes

- class [es.deusto.client.windows.ChangePass](#)  
*This class is the one used to create a window to change the user password.*

## Packages

- package [es.deusto.client.windows](#)

## 8.4 src/main/java/es/deusto/client/windows/CodeVerification.java File Reference

### Classes

- class [es.deusto.client.windows.CodeVerification](#)

## Packages

- package [es.deusto.client.windows](#)

## 8.5 src/main/java/es/deusto/client/windows/CreateEvent.java File Reference

### Classes

- class [es.deusto.client.windows.CreateEvent](#)

*This class is the one used to create a window that it allows the user to create a new event.*

## Packages

- package [es.deusto.client.windows](#)

## 8.6 src/main/java/es/deusto/client/windows/CreateOrganizerWindow.java File Reference

### Classes

- class [es.deusto.client.windows.CreateOrganizerWindow](#)

*This class is the one used to create a window that allows to create a new organizer.*

## Packages

- package [es.deusto.client.windows](#)

## 8.7 src/main/java/es/deusto/client/windows/CreateUserWindow.java File Reference

### Classes

- class [es.deusto.client.windows.CreateUserWindow](#)  
*This class is the one used to create a window that allows to create a new user.*

### Packages

- package [es.deusto.client.windows](#)

## 8.8 src/main/java/es/deusto/client/windows/EventListItem.java File Reference

### Classes

- class [es.deusto.client.windows.EventListItem](#)  
*This class is the one used to create a window that allows the user to have an event list.*

### Packages

- package [es.deusto.client.windows](#)

## 8.9 src/main/java/es/deusto/client/windows/EventOrganizerWindow.java File Reference

### Classes

- class [es.deusto.client.windows.EventOrganizerWindow](#)  
*This class is the one used to create a window that allows to an organizer to change or administrate one of his events.*

### Packages

- package [es.deusto.client.windows](#)

## 8.10 src/main/java/es/deusto/client/windows/EventWindow.java File Reference

### Classes

- class [es.deusto.client.windows.EventWindow](#)  
*This class is the one used to create a window that allows to display all the information of one event.*

## Packages

- package [es.deusto.client.windows](#)

## 8.11 [src/main/java/es/deusto/client/windows/ForgottenPassword.java](#) File Reference

### Classes

- class [es.deusto.client.windows.ForgottenPassword](#)  
*This class is the one used to create a window that allows a user to log in when he forgets the password.*

## Packages

- package [es.deusto.client.windows](#)

## 8.12 [src/main/java/es/deusto/client/windows/LanguageManager.java](#) File Reference

### Classes

- class [es.deusto.client.windows.LanguageManager](#)  
*This class is used to manage the different languages supported by the application.*

## Packages

- package [es.deusto.client.windows](#)

## 8.13 [src/main/java/es/deusto/client/windows/LogInWindow.java](#) File Reference

### Classes

- class [es.deusto.client.windows.LogInWindow](#)  
*This class is the one used to create a window that allows a user/organizer to log in to the application.*

## Packages

- package [es.deusto.client.windows](#)



## 8.14 src/main/java/es/deusto/client/windows/OrganizerEvents.java File Reference

### Classes

- class [es.deusto.client.windows.OrganizerEvents](#)

*This class is the one used to create a window that allows an organizer to have all of his events.*

### Packages

- package [es.deusto.client.windows](#)

## 8.15 src/main/java/es/deusto/client/windows/OrganizerHome.java File Reference

### Classes

- class [es.deusto.client.windows.OrganizerHome](#)

*This class is the one used to create a window that allows an organizer to have a home window where he can access to all of the functionalities of the application.*

### Packages

- package [es.deusto.client.windows](#)

## 8.16 src/main/java/es/deusto/client/windows/PostCreationWindow.java File Reference

### Classes

- class [es.deusto.client.windows.PostCreationWindow](#)

*This class is the one used to create a window that allows an organizer to have like a double check of the event he has just created.*

### Packages

- package [es.deusto.client.windows](#)

## 8.17 src/main/java/es/deusto/client/windows/Profile.java File Reference

### Classes

- class [es.deusto.client.windows.Profile](#)

*This class is the one used to create a window that allows a user/organizer to have a profile with all of his information.*

## Packages

- package [es.deusto.client.windows](#)

## 8.18 src/main/java/es/deusto/client/windows/SavedEventsWindow.java File Reference

### Classes

- class [es.deusto.client.windows.SavedEventsWindow](#)

*This class is the one used to create a window that allows to scroll, search and choose from all of the events stored in the database.*

## Packages

- package [es.deusto.client.windows](#)

## 8.19 src/main/java/es/deusto/client/windows/UserEventsWindow.java File Reference

### Classes

- class [es.deusto.client.windows.UserEventsWindow](#)

*This class is the one used to create a window that allows a user to scroll, search and select from a list of events he previously has chosen from.*

## Packages

- package [es.deusto.client.windows](#)

## 8.20 src/main/java/es/deusto/serialization/EventInfo.java File Reference

### Classes

- class [es.deusto.serialization.EventInfo](#)

*DTO equivalent class of the Event class from the server.*

## Packages

- package [es.deusto.serialization](#)

## 8.21 src/main/java/es/deusto/serialization/GenericUserInfo.java File Reference

### Classes

- class [es.deusto.serialization.GenericUserInfo](#)

*This class works as a Tag for the Classes related to the User and Organizer.*

### Packages

- package [es.deusto.serialization](#)

## 8.22 src/main/java/es/deusto/serialization/LoginAttempt.java File Reference

### Classes

- class [es.deusto.serialization.LoginAttempt](#)

*This class encapsulates all the information required when a user (Either normal User or Organizer) logs in.*

### Packages

- package [es.deusto.serialization](#)

## 8.23 src/main/java/es/deusto/serialization/OrganizerInfo.java File Reference

### Classes

- class [es.deusto.serialization.OrganizerInfo](#)

*DTO equivalent class of the Organizer class from the server.*

### Packages

- package [es.deusto.serialization](#)

## 8.24 src/main/java/es/deusto/serialization/PostInfo.java File Reference

### Classes

- class [es.deusto.serialization.PostInfo](#)

*DTO equivalent class of the Post class from the server.*

## Packages

- package [es.deusto.serialization](#)

## 8.25 src/main/java/es/deusto/serialization/SignupAttempt.java File Reference

### Classes

- class [es.deusto.serialization.SignupAttempt](#)  
*This class encapsulates all the information required when a user (Either normal User or Organizer) signs up.*

## Packages

- package [es.deusto.serialization](#)

## 8.26 src/main/java/es/deusto/serialization/TopicInfo.java File Reference

### Classes

- class [es.deusto.serialization.TopicInfo](#)  
*DTO equivalent class of the Topic class from the server.*

## Packages

- package [es.deusto.serialization](#)

## 8.27 src/main/java/es/deusto/serialization/UserInfo.java File Reference

### Classes

- class [es.deusto.serialization.UserInfo](#)  
*DTO equivalent class of the User class from the server.*

## Packages

- package [es.deusto.serialization](#)

## 8.28 src/main/java/es/deusto/server/dao/DAOFactory.java File Reference

### Classes

- class [es.deusto.server.dao.DAOFactory](#)  
*This class is used to create a DAO factory that is used by all of the other DAO classes, to create them or to close them.*

## Packages

- package [es.deusto.server.dao](#)

## 8.29 src/main/java/es/deusto/server/dao/EventDAO.java File Reference

### Classes

- class [es.deusto.server.dao.EventDAO](#)  
*This classes is the one used to create an event DAO.*

## Packages

- package [es.deusto.server.dao](#)

## 8.30 src/main/java/es/deusto/server/dao/OrganizerDAO.java File Reference

### Classes

- class [es.deusto.server.dao.OrganizerDAO](#)  
*This classes is the one used to create an organizer DAO.*

## Packages

- package [es.deusto.server.dao](#)

## 8.31 src/main/java/es/deusto/server/dao/PostDAO.java File Reference

### Classes

- class [es.deusto.server.dao.PostDAO](#)  
*This classes is the one used to create a DAO for a post.*

## Packages

- package [es.deusto.server.dao](#)

## 8.32 src/main/java/es/deusto/server/dao/TopicDAO.java File Reference

### Classes

- class [es.deusto.server.dao.TopicDAO](#)  
*This classes is the one used to create a DAO a topic.*

## Packages

- package [es.deusto.server.dao](#)

## 8.33 src/main/java/es/deusto/server/dao/UserDAO.java File Reference

### Classes

- class [es.deusto.server.dao.UserDAO](#)  
*This classes is the one used to create an user DAO.*

## Packages

- package [es.deusto.server.dao](#)

## 8.34 src/main/java/es/deusto/server/data/Event.java File Reference

### Classes

- class [es.deusto.server.data.Event](#)  
*This classes is the one that defines an event and it is used to create one.*

## Packages

- package [es.deusto.server.data](#)

## 8.35 src/main/java/es/deusto/server/data/GenericUser.java File Reference

### Classes

- class [es.deusto.server.data.GenericUser](#)  
*This classes is the father class for normal user and organizer.*

## Packages

- package [es.deusto.server.data](#)

## 8.36 src/main/java/es/deusto/server/data/Organizer.java File Reference

### Classes

- class [es.deusto.server.data.Organizer](#)  
*This classes is the one that defines an organizer and it is used to create one.*

## Packages

- package [es.deusto.server.data](#)

## 8.37 src/main/java/es/deusto/server/data/Post.java File Reference

### Classes

- class [es.deusto.server.data.Post](#)

*This classes is the one that defines a post and it is used to create one.*

## Packages

- package [es.deusto.server.data](#)

## 8.38 src/main/java/es/deusto/server/data/Topic.java File Reference

### Classes

- class [es.deusto.server.data.Topic](#)

*This classes is the one that defines a topic and it is used to create one.*

## Packages

- package [es.deusto.server.data](#)

## 8.39 src/main/java/es/deusto/server/data/User.java File Reference

### Classes

- class [es.deusto.server.data.User](#)

*This classes is the one that defines a normal user and it is used to create one.*

## Packages

- package [es.deusto.server.data](#)

## 8.40 src/main/java/es/deusto/server/server/AppService.java File Reference

### Classes

- class [es.deusto.server.server.AppService](#)

## Packages

- package [es.deusto.server.server](#)

## 8.41 src/main/java/es/deusto/server/server/Server.java File Reference

### Classes

- class [es.deusto.server.server.Server](#)  
*This class is the one that receives the requests from the client.*

## Packages

- package [es.deusto.server.server](#)

## 8.42 src/test/java/es/deusto/junit/client/ControllerTest.java File Reference

### Classes

- class [es.deusto.junit.client.ControllerTest](#)  
*This classes is the one that tests the controller component to see if it is working as intended.*

## Packages

- package [es.deusto.junit.client](#)

## 8.43 src/test/java/es/deusto/junit/serialization/SerializationTest.java File Reference

### Classes

- class [es.deusto.junit.serialization.SerializationTest](#)  
*This classes is the one that tests funcionalities such as creating an event, creating a user, orginzer, crea ting posts and topics.*

## Packages

- package [es.deusto.junit.serialization](#)

## 8.44 src/test/java/es/deusto/junit/server/ServerTest.java File Reference

### Classes

- class [es.deusto.junit.server.ServerTest](#)  
*This classes is used to create several tests to see if the server is working as intended.*

## Packages

- package [es.deusto.junit.server](#)



# Index

- actionPerformed
  - es.deusto.client.windows.CreateEvent, [41](#)
  - es.deusto.client.windows.CreateOrganizerWindow, [42](#)
  - es.deusto.client.windows.CreateUserWindow, [43](#)
- addCreatedEvent
  - es.deusto.server.data.Organizer, [69](#)
- addEvent
  - es.deusto.server.data.User, [107](#)
- addInterest
  - es.deusto.serialization.UserInfo, [113](#)
  - es.deusto.server.data.User, [108](#)
- addInterested
  - es.deusto.serialization.EventInfo, [54](#)
  - es.deusto.server.data.Event, [47](#)
- AppService
  - es.deusto.server.server.AppService, [20](#)
- attemptEventDelete
  - es.deusto.client.controller.Controller, [30](#)
- attemptNormalLogin
  - es.deusto.client.controller.Controller, [31](#)
  - es.deusto.server.server.AppService, [20](#)
  - es.deusto.server.server.Server, [89](#)
- attemptNormalLoginOrganizer
  - es.deusto.client.controller.Controller, [32](#)
- attemptNormalSignup
  - es.deusto.client.controller.Controller, [32](#)
  - es.deusto.server.server.AppService, [20](#)
  - es.deusto.server.server.Server, [89](#)
- attemptNormalUpdate
  - es.deusto.client.controller.Controller, [33](#)
  - es.deusto.server.server.AppService, [21](#)
  - es.deusto.server.server.Server, [90](#)
- attemptOrganizerDelete
  - es.deusto.client.controller.Controller, [33](#)
  - es.deusto.server.server.Server, [90](#)
- attemptOrganizerLogin
  - es.deusto.server.server.AppService, [21](#)
  - es.deusto.server.server.Server, [91](#)
- attemptOrganizerSignup
  - es.deusto.client.controller.Controller, [34](#)
  - es.deusto.server.server.AppService, [22](#)
  - es.deusto.server.server.Server, [91](#)
- attemptOrganizerUpdate
  - es.deusto.client.controller.Controller, [34](#)
  - es.deusto.server.server.AppService, [22](#)
  - es.deusto.server.server.Server, [92](#)
- attemptPasswordRecovery
  - es.deusto.client.controller.Controller, [35](#)
- attemptUserDelete
  - es.deusto.client.controller.Controller, [35](#)
  - es.deusto.server.server.Server, [92](#)
- bCreate
  - es.deusto.client.windows.LogInWindow, [67](#)
- beforeClass
  - es.deusto.junit.serialization.SerializationTest, [85](#)
- buildOrganizer
  - es.deusto.serialization.SignupAttempt, [99](#)
- buildUser
  - es.deusto.serialization.SignupAttempt, [99](#)
- cBoxAdmin
  - es.deusto.client.windows.LogInWindow, [67](#)
- ChangePass
  - es.deusto.client.windows.ChangePass, [27](#)
- close
  - es.deusto.server.dao.EventDAO, [50](#)
  - es.deusto.server.dao.OrganizerDAO, [71](#)
  - es.deusto.server.dao.PostDAO, [80](#)
  - es.deusto.server.dao.TopicDAO, [104](#)
  - es.deusto.server.dao.UserDAO, [110](#)
- closeDAO
  - es.deusto.server.dao.DAOFactory, [44](#)
- CodeVerification
  - es.deusto.client.windows.CodeVerification, [28](#)
- configure
  - es.deusto.junit.client.ControllerTest, [38](#)
- Controller
  - es.deusto.client.controller.Controller, [30](#)
- CreateEvent
  - es.deusto.client.windows.CreateEvent, [41](#)
- createEvent
  - es.deusto.client.controller.Controller, [35](#)
  - es.deusto.server.server.AppService, [23](#)
  - es.deusto.server.server.Server, [93](#)
- createEventDAO
  - es.deusto.server.dao.DAOFactory, [44](#)
- createOrganizerDAO
  - es.deusto.server.dao.DAOFactory, [45](#)
- CreateOrganizerWindow
  - es.deusto.client.windows.CreateOrganizerWindow, [42](#)
- createPost
  - es.deusto.client.controller.Controller, [36](#)
  - es.deusto.server.server.AppService, [23](#)
  - es.deusto.server.server.Server, [93](#)
- createPostDAO
  - es.deusto.server.dao.DAOFactory, [45](#)

- createTopicDAO
  - es.deusto.server.dao.DAOFactory, 45
- createUserDAO
  - es.deusto.server.dao.DAOFactory, 45
- CreateUserWindow
  - es.deusto.client.windows.CreateUserWindow, 43
- deleteEvent
  - es.deusto.server.dao.EventDAO, 51
  - es.deusto.server.server.AppService, 24
  - es.deusto.server.server.Server, 94
- deleteOrganizer
  - es.deusto.server.dao.OrganizerDAO, 71
  - es.deusto.server.server.AppService, 24
- deletePost
  - es.deusto.server.dao.PostDAO, 80
- deleteTopic
  - es.deusto.server.dao.TopicDAO, 104
- deleteUser
  - es.deusto.server.dao.UserDAO, 110
  - es.deusto.server.server.AppService, 25
- es, 13
- es.deusto, 13
- es.deusto.client, 13
- es.deusto.client.controller, 13
- es.deusto.client.controller.Controller, 28
  - attemptEventDelete, 30
  - attemptNormalLogin, 31
  - attemptNormalLoginOrganizer, 32
  - attemptNormalSignup, 32
  - attemptNormalUpdate, 33
  - attemptOrganizerDelete, 33
  - attemptOrganizerSignup, 34
  - attemptOrganizerUpdate, 34
  - attemptPasswordRecovery, 35
  - attemptUserDelete, 35
  - Controller, 30
  - createEvent, 35
  - createPost, 36
  - getLanguageManager, 36
  - getOrganize, 37
  - getRecommendations, 37
  - getUser, 37
  - setOrganizer, 37
  - setUser, 37
- es.deusto.client.main, 14
- es.deusto.client.main.Main, 68
  - main, 68
- es.deusto.client.windows, 14
- es.deusto.client.windows.ChangePass, 27
  - ChangePass, 27
- es.deusto.client.windows.CodeVerification, 28
  - CodeVerification, 28
- es.deusto.client.windows.CreateEvent, 40
  - actionPerformed, 41
  - CreateEvent, 41
- es.deusto.client.windows.CreateOrganizerWindow, 41
  - actionPerformed, 42
  - CreateOrganizerWindow, 42
- es.deusto.client.windows.CreateUserWindow, 42
  - actionPerformed, 43
  - CreateUserWindow, 43
- es.deusto.client.windows.EventListItem, 57
  - EventListItem, 57
- es.deusto.client.windows.EventOrganizerWindow, 58
  - EventOrganizerWindow, 58
- es.deusto.client.windows.EventWindow, 58
  - EventWindow, 59
- es.deusto.client.windows.ForgottenPassword, 59
  - ForgottenPassword, 60
- es.deusto.client.windows.LanguageManager, 62
  - getLanguage, 63
  - getString, 63
  - LanguageManager, 62, 63
  - setLanguage, 63
- es.deusto.client.windows.LogInWindow, 66
  - bCreate, 67
  - cBoxAdmin, 67
  - id, 67
  - LogInWindow, 67
- es.deusto.client.windows.OrganizerEvents, 72
  - OrganizerEvents, 72
- es.deusto.client.windows.OrganizerHome, 73
  - OrganizerHome, 73
- es.deusto.client.windows.PostCreationWindow, 78
  - PostCreationWindow, 79
- es.deusto.client.windows.Profile, 83
  - Profile, 84
- es.deusto.client.windows.SavedEventsWindow, 84
  - SavedEventsWindow, 84
- es.deusto.client.windows.UserEventsWindow, 111
  - UserEventsWindow, 112
- es.deusto.junit, 15
- es.deusto.junit.client, 15
- es.deusto.junit.client.ControllerTest, 38
  - configure, 38
  - rule, 40
  - setUpChild, 38
  - startUp, 39
  - suite, 39
  - tearDownChild, 39
  - testCreateDeleteEvent, 39
  - testNormalLogin, 39
  - testNormalSignupAndDelete, 39
  - testOrganizerLogin, 39
  - testOrganizerSignupAndDelete, 39
  - testUserUpdate, 40
- es.deusto.junit.serialization, 15
- es.deusto.junit.serialization.SerializationTest, 85
  - beforeClass, 85
  - startUp, 86
  - suite, 86
  - testEventInfoFromEvent, 86
  - testEventInfoSetters, 86
  - testOrganizerInfoFromOrganizer, 86
  - testOrganizerInfoSetters, 86

- testPostInfoFromPost, 86
- testPostInfoSetters, 86
- testSignupAttemptOrganizer, 87
- testSignupAttemptUser, 87
- testUserInfoFromUser, 87
- testUserInfoSetters, 87
- testUserInfoStringConstructor, 87
- es.deusto.junit.server, 15
- es.deusto.junit.server.ServerTest, 95
  - setUp, 96
  - startUp, 96
  - suite, 96
  - tearDown, 96
  - testCreateEvent, 97
  - testCreatePost, 97
  - testNormalSignup, 97
  - testNormalUpdate, 97
  - testOrganizerLogin, 97
  - testOrganizerSignup, 97
  - testOrganizerUpdate, 97
- es.deusto.serialization, 15
- es.deusto.serialization.EventInfo, 53
  - addInterested, 54
  - EventInfo, 54
  - getCity, 54
  - getCountry, 54
  - getDate, 54
  - getDescription, 54
  - getInterested, 55
  - getName, 55
  - getOrganizerEmail, 55
  - getPosts, 55
  - getTopic, 55
  - reduceInterested, 55
  - setCity, 55
  - setCountry, 55
  - setDate, 56
  - setDescription, 56
  - setInterested, 56
  - setName, 56
  - setOrganizerEmail, 56
  - setPosts, 56
  - setTopic, 56
- es.deusto.serialization.GenericUserInfo, 62
- es.deusto.serialization.LoginAttempt, 64
  - getEmail, 65
  - getPassword, 65
  - isOrganizer, 65
  - LoginAttempt, 64, 65
  - setEmail, 65
  - setOrganizer, 66
  - setPassword, 66
  - toString, 66
- es.deusto.serialization.OrganizerInfo, 73
  - getCreatedEvents, 74
  - getEmail, 74
  - getName, 75
  - getOrganization, 75
  - OrganizerInfo, 74
  - setCreatedEvents, 75
  - setEmail, 75
  - setName, 75
  - setOrganization, 75
- es.deusto.serialization.PostInfo, 81
  - getDate, 82
  - getDescription, 82
  - getEventName, 82
  - getOrganizerEmail, 82
  - getTitle, 82
  - PostInfo, 81
  - setDate, 82
  - setDescription, 82
  - setEventName, 83
  - setOrganizerEmail, 83
  - setTitle, 83
- es.deusto.serialization.SignupAttempt, 98
  - buildOrganizer, 99
  - buildUser, 99
  - getCity, 99
  - getCountry, 99
  - getEmail, 100
  - getInterests, 100
  - getName, 100
  - getOrganization, 100
  - getPassword, 100
  - getSavedEvents, 100
  - setCity, 100
  - setCountry, 100
  - setEmail, 101
  - setInterests, 101
  - setName, 101
  - setOrganization, 101
  - setPassword, 101
  - setSaveEvents, 101
  - SignupAttempt, 98, 99
  - toString, 101
- es.deusto.serialization.TopicInfo, 105
  - getName, 106
  - parseTopic, 106
  - setName, 106
  - TopicInfo, 105, 106
  - toString, 106
- es.deusto.serialization.UserInfo, 112
  - addInterest, 113
  - getCity, 113
  - getCountry, 114
  - getEmail, 114
  - getInterests, 114
  - getName, 114
  - getSavedEvents, 114
  - setCity, 114
  - setCountry, 114
  - setEmail, 115
  - setInterests, 115
  - setName, 115
  - setSavedEvents, 115

- UserInfo, 113
- es.deusto.server, 16
- es.deusto.server.dao, 16
- es.deusto.server.dao.DAOFactory, 43
  - closeDAO, 44
  - createEventDAO, 44
  - createOrganizerDAO, 45
  - createPostDAO, 45
  - createTopicDAO, 45
  - createUserDAO, 45
  - getInstance, 45
- es.deusto.server.dao.EventDAO, 50
  - close, 50
  - deleteEvent, 51
  - EventDAO, 50
  - getEvents, 51
  - getEventsByOrganizer, 51
  - getEventsbyUser, 52
  - getPersistenceManager, 52
  - storeEvent, 52
  - updateEvent, 52
- es.deusto.server.dao.OrganizerDAO, 70
  - close, 71
  - deleteOrganizer, 71
  - getOrganizer, 71
  - getPersistenceManager, 71
  - OrganizerDAO, 71
  - storeOrganizer, 71
  - updateOrganizer, 71
- es.deusto.server.dao.PostDAO, 79
  - close, 80
  - deletePost, 80
  - getPersistenceManager, 80
  - getPostsByEvent, 80
  - PostDAO, 79
  - storePost, 80
  - updatePost, 80
- es.deusto.server.dao.TopicDAO, 103
  - close, 104
  - deleteTopic, 104
  - getPersistenceManager, 104
  - getTopic, 104
  - storeTopic, 104
  - TopicDAO, 104
  - updateTopic, 104
- es.deusto.server.dao.UserDAO, 109
  - close, 110
  - deleteUser, 110
  - getAllUsers, 110
  - getPersistenceManager, 110
  - getUser, 110
  - storeUser, 111
  - updateUser, 111
  - UserDAO, 110
- es.deusto.server.data, 16
- es.deusto.server.data.Event, 45
  - addInterested, 47
  - Event, 46
  - getCity, 47
  - getCountry, 47
  - getDate, 47
  - getDescription, 47
  - getInterested, 47
  - getName, 47
  - getOrganizer, 47
  - getPosts, 48
  - getTopic, 48
  - reduceInterested, 48
  - setCity, 48
  - setCountry, 48
  - setDate, 48
  - setDescription, 48
  - setInterested, 49
  - setName, 49
  - setOrganizer, 49
  - setPosts, 49
  - setTopic, 49
  - toString, 49
- es.deusto.server.data.GenericUser, 60
  - getEmail, 60
  - getName, 61
  - getPassword, 61
  - setEmail, 61
  - setName, 61
  - setPassword, 61
  - toString, 61
- es.deusto.server.data.Organizer, 68
  - addCreatedEvent, 69
  - getCreatedEvents, 69
  - getOrganization, 69
  - Organizer, 69
  - setCreatedEvents, 69
  - setOrganization, 70
  - toString, 70
- es.deusto.server.data.Post, 76
  - getDate, 76
  - getDescription, 77
  - getEventName, 77
  - getOrganizerEmail, 77
  - getTitle, 77
  - Post, 76
  - setDate, 77
  - setDescription, 77
  - setEventName, 77
  - setOrganizerEmail, 78
  - setTitle, 78
- es.deusto.server.data.Topic, 102
  - getName, 102
  - setName, 103
  - Topic, 102
  - toString, 103
- es.deusto.server.data.User, 107
  - addEvent, 107
  - addInterest, 108
  - getCity, 108
  - getCountry, 108

- getInterests, 108
- getSavedEvents, 108
- setCity, 108
- setCountry, 108
- setInterests, 109
- setSavedEvents, 109
- toString, 109
- es.deusto.server.server, 17
- es.deusto.server.server.AppService, 19
  - AppService, 20
  - attemptNormalLogin, 20
  - attemptNormalSignup, 20
  - attemptNormalUpdate, 21
  - attemptOrganizerLogin, 21
  - attemptOrganizerSignup, 22
  - attemptOrganizerUpdate, 22
  - createEvent, 23
  - createPost, 23
  - deleteEvent, 24
  - deleteOrganizer, 24
  - deleteUser, 25
  - getRecommendedEvents, 25
  - hello, 26
  - recoverPassword, 26
  - updateUser, 26
- es.deusto.server.server.Server, 87
  - attemptNormalLogin, 89
  - attemptNormalSignup, 89
  - attemptNormalUpdate, 90
  - attemptOrganizerDelete, 90
  - attemptOrganizerLogin, 91
  - attemptOrganizerSignup, 91
  - attemptOrganizerUpdate, 92
  - attemptUserDelete, 92
  - createEvent, 93
  - createPost, 93
  - deleteEvent, 94
  - getRecommendedEvents, 94
  - recoverPassword, 94
  - sayHello, 95
  - Server, 88
- Event
  - es.deusto.server.data.Event, 46
- EventDAO
  - es.deusto.server.dao.EventDAO, 50
- EventInfo
  - es.deusto.serialization.EventInfo, 54
- EventListItem
  - es.deusto.client.windows.EventListItem, 57
- EventOrganizerWindow
  - es.deusto.client.windows.EventOrganizerWindow, 58
- EventWindow
  - es.deusto.client.windows.EventWindow, 59
- ForgottenPassword
  - es.deusto.client.windows.ForgottenPassword, 60
- getAllUsers
  - es.deusto.server.dao.UserDAO, 110
- getCity
  - es.deusto.serialization.EventInfo, 54
  - es.deusto.serialization.SignupAttempt, 99
  - es.deusto.serialization.UserInfo, 113
  - es.deusto.server.data.Event, 47
  - es.deusto.server.data.User, 108
- getCountry
  - es.deusto.serialization.EventInfo, 54
  - es.deusto.serialization.SignupAttempt, 99
  - es.deusto.serialization.UserInfo, 114
  - es.deusto.server.data.Event, 47
  - es.deusto.server.data.User, 108
- getCreatedEvents
  - es.deusto.serialization.OrganizerInfo, 74
  - es.deusto.server.data.Organizer, 69
- getDate
  - es.deusto.serialization.EventInfo, 54
  - es.deusto.serialization.PostInfo, 82
  - es.deusto.server.data.Event, 47
  - es.deusto.server.data.Post, 76
- getDescription
  - es.deusto.serialization.EventInfo, 54
  - es.deusto.serialization.PostInfo, 82
  - es.deusto.server.data.Event, 47
  - es.deusto.server.data.Post, 77
- getEmail
  - es.deusto.serialization.LoginAttempt, 65
  - es.deusto.serialization.OrganizerInfo, 74
  - es.deusto.serialization.SignupAttempt, 100
  - es.deusto.serialization.UserInfo, 114
  - es.deusto.server.data.GenericUser, 60
- getEventName
  - es.deusto.serialization.PostInfo, 82
  - es.deusto.server.data.Post, 77
- getEvents
  - es.deusto.server.dao.EventDAO, 51
- getEventsByOrganizer
  - es.deusto.server.dao.EventDAO, 51
- getEventsbyUser
  - es.deusto.server.dao.EventDAO, 52
- getInstance
  - es.deusto.server.dao.DAOFactory, 45
- getInterested
  - es.deusto.serialization.EventInfo, 55
  - es.deusto.server.data.Event, 47
- getInterests
  - es.deusto.serialization.SignupAttempt, 100
  - es.deusto.serialization.UserInfo, 114
  - es.deusto.server.data.User, 108
- getLanguage
  - es.deusto.client.windows.LanguageManager, 63
- getLanguageManager
  - es.deusto.client.controller.Controller, 36
- getName
  - es.deusto.serialization.EventInfo, 55
  - es.deusto.serialization.OrganizerInfo, 75
  - es.deusto.serialization.SignupAttempt, 100

- es.deusto.serialization.TopicInfo, 106
  - es.deusto.serialization.UserInfo, 114
  - es.deusto.server.data.Event, 47
  - es.deusto.server.data.GenericUser, 61
  - es.deusto.server.data.Topic, 102
- getOrganization
  - es.deusto.serialization.OrganizerInfo, 75
  - es.deusto.serialization.SignupAttempt, 100
  - es.deusto.server.data.Organizer, 69
- getOrganize
  - es.deusto.client.controller.Controller, 37
- getOrganizer
  - es.deusto.server.dao.OrganizerDAO, 71
  - es.deusto.server.data.Event, 47
- getOrganizerEmail
  - es.deusto.serialization.EventInfo, 55
  - es.deusto.serialization.PostInfo, 82
  - es.deusto.server.data.Post, 77
- getPassword
  - es.deusto.serialization.LoginAttempt, 65
  - es.deusto.serialization.SignupAttempt, 100
  - es.deusto.server.data.GenericUser, 61
- getPersistenceManager
  - es.deusto.server.dao.EventDAO, 52
  - es.deusto.server.dao.OrganizerDAO, 71
  - es.deusto.server.dao.PostDAO, 80
  - es.deusto.server.dao.TopicDAO, 104
  - es.deusto.server.dao.UserDAO, 110
- getPosts
  - es.deusto.serialization.EventInfo, 55
  - es.deusto.server.data.Event, 48
- getPostsByEvent
  - es.deusto.server.dao.PostDAO, 80
- getRecommendations
  - es.deusto.client.controller.Controller, 37
- getRecommendedEvents
  - es.deusto.server.server.AppService, 25
  - es.deusto.server.server.Server, 94
- getSavedEvents
  - es.deusto.serialization.SignupAttempt, 100
  - es.deusto.serialization.UserInfo, 114
  - es.deusto.server.data.User, 108
- getString
  - es.deusto.client.windows.LanguageManager, 63
- getTitle
  - es.deusto.serialization.PostInfo, 82
  - es.deusto.server.data.Post, 77
- getTopic
  - es.deusto.serialization.EventInfo, 55
  - es.deusto.server.dao.TopicDAO, 104
  - es.deusto.server.data.Event, 48
- getUser
  - es.deusto.client.controller.Controller, 37
  - es.deusto.server.dao.UserDAO, 110
- hello
  - es.deusto.server.server.AppService, 26
- id
  - es.deusto.client.windows.LogInWindow, 67
- isOrganizer
  - es.deusto.serialization.LoginAttempt, 65
- LanguageManager
  - es.deusto.client.windows.LanguageManager, 62, 63
- LoginAttempt
  - es.deusto.serialization.LoginAttempt, 64, 65
- LogInWindow
  - es.deusto.client.windows.LogInWindow, 67
- main
  - es.deusto.client.main.Main, 68
- Organizer
  - es.deusto.server.data.Organizer, 69
- OrganizerDAO
  - es.deusto.server.dao.OrganizerDAO, 71
- OrganizerEvents
  - es.deusto.client.windows.OrganizerEvents, 72
- OrganizerHome
  - es.deusto.client.windows.OrganizerHome, 73
- OrganizerInfo
  - es.deusto.serialization.OrganizerInfo, 74
- parseTopic
  - es.deusto.serialization.TopicInfo, 106
- Post
  - es.deusto.server.data.Post, 76
- PostCreationWindow
  - es.deusto.client.windows.PostCreationWindow, 79
- PostDAO
  - es.deusto.server.dao.PostDAO, 79
- PostInfo
  - es.deusto.serialization.PostInfo, 81
- Profile
  - es.deusto.client.windows.Profile, 84
- recoverPassword
  - es.deusto.server.server.AppService, 26
  - es.deusto.server.server.Server, 94
- reduceInterested
  - es.deusto.serialization.EventInfo, 55
  - es.deusto.server.data.Event, 48
- rule
  - es.deusto.junit.client.ControllerTest, 40
- SavedEventsWindow
  - es.deusto.client.windows.SavedEventsWindow, 84
- sayHello
  - es.deusto.server.server.Server, 95
- Server
  - es.deusto.server.server.Server, 88
- setCity
  - es.deusto.serialization.EventInfo, 55
  - es.deusto.serialization.SignupAttempt, 100
  - es.deusto.serialization.UserInfo, 114
  - es.deusto.server.data.Event, 48
  - es.deusto.server.data.User, 108



- setCountry
  - es.deusto.serialization.EventInfo, 55
  - es.deusto.serialization.SignupAttempt, 100
  - es.deusto.serialization.UserInfo, 114
  - es.deusto.server.data.Event, 48
  - es.deusto.server.data.User, 108
- setCreatedEvents
  - es.deusto.serialization.OrganizerInfo, 75
  - es.deusto.server.data.Organizer, 69
- setDate
  - es.deusto.serialization.EventInfo, 56
  - es.deusto.serialization.PostInfo, 82
  - es.deusto.server.data.Event, 48
  - es.deusto.server.data.Post, 77
- setDescription
  - es.deusto.serialization.EventInfo, 56
  - es.deusto.serialization.PostInfo, 82
  - es.deusto.server.data.Event, 48
  - es.deusto.server.data.Post, 77
- setEmail
  - es.deusto.serialization.LoginAttempt, 65
  - es.deusto.serialization.OrganizerInfo, 75
  - es.deusto.serialization.SignupAttempt, 101
  - es.deusto.serialization.UserInfo, 115
  - es.deusto.server.data.GenericUser, 61
- setEventName
  - es.deusto.serialization.PostInfo, 83
  - es.deusto.server.data.Post, 77
- setInterested
  - es.deusto.serialization.EventInfo, 56
  - es.deusto.server.data.Event, 49
- setInterests
  - es.deusto.serialization.SignupAttempt, 101
  - es.deusto.serialization.UserInfo, 115
  - es.deusto.server.data.User, 109
- setLanguage
  - es.deusto.client.windows.LanguageManager, 63
- setName
  - es.deusto.serialization.EventInfo, 56
  - es.deusto.serialization.OrganizerInfo, 75
  - es.deusto.serialization.SignupAttempt, 101
  - es.deusto.serialization.TopicInfo, 106
  - es.deusto.serialization.UserInfo, 115
  - es.deusto.server.data.Event, 49
  - es.deusto.server.data.GenericUser, 61
  - es.deusto.server.data.Topic, 103
- setOrganization
  - es.deusto.serialization.OrganizerInfo, 75
  - es.deusto.serialization.SignupAttempt, 101
  - es.deusto.server.data.Organizer, 70
- setOrganizer
  - es.deusto.client.controller.Controller, 37
  - es.deusto.serialization.LoginAttempt, 66
  - es.deusto.server.data.Event, 49
- setOrganizerEmail
  - es.deusto.serialization.EventInfo, 56
  - es.deusto.serialization.PostInfo, 83
  - es.deusto.server.data.Post, 78
- setPassword
  - es.deusto.serialization.LoginAttempt, 66
  - es.deusto.serialization.SignupAttempt, 101
  - es.deusto.server.data.GenericUser, 61
- setPosts
  - es.deusto.serialization.EventInfo, 56
  - es.deusto.server.data.Event, 49
- setSavedEvents
  - es.deusto.serialization.UserInfo, 115
  - es.deusto.server.data.User, 109
- setSaveEvents
  - es.deusto.serialization.SignupAttempt, 101
- setTitle
  - es.deusto.serialization.PostInfo, 83
  - es.deusto.server.data.Post, 78
- setTopic
  - es.deusto.serialization.EventInfo, 56
  - es.deusto.server.data.Event, 49
- setUp
  - es.deusto.junit.server.ServerTest, 96
- setUpChild
  - es.deusto.junit.client.ControllerTest, 38
- setUser
  - es.deusto.client.controller.Controller, 37
- SignupAttempt
  - es.deusto.serialization.SignupAttempt, 98, 99
- src/main/java/es/deusto/client/controller/Controller.java, 117
- src/main/java/es/deusto/client/main/Main.java, 117
- src/main/java/es/deusto/client/windows/ChangePass.java, 117
- src/main/java/es/deusto/client/windows/CodeVerification.java, 118
- src/main/java/es/deusto/client/windows/CreateEvent.java, 118
- src/main/java/es/deusto/client/windows/CreateOrganizerWindow.java, 118
- src/main/java/es/deusto/client/windows/CreateUserWindow.java, 119
- src/main/java/es/deusto/client/windows/EventListItem.java, 119
- src/main/java/es/deusto/client/windows/EventOrganizerWindow.java, 119
- src/main/java/es/deusto/client/windows/EventWindow.java, 119
- src/main/java/es/deusto/client/windows/ForgottenPassword.java, 120
- src/main/java/es/deusto/client/windows/LanguageManager.java, 120
- src/main/java/es/deusto/client/windows/LogInWindow.java, 120
- src/main/java/es/deusto/client/windows/OrganizerEvents.java, 121
- src/main/java/es/deusto/client/windows/OrganizerHome.java, 121
- src/main/java/es/deusto/client/windows/PostCreationWindow.java, 121
- src/main/java/es/deusto/client/windows/Profile.java, 121

[src/main/java/es/deusto/client/windows/SavedEventsWindow.java](#), 122  
[src/main/java/es/deusto/client/windows/UserEventsWindow.java](#), 122  
[src/main/java/es/deusto/serialization/EventInfo.java](#), 122  
[src/main/java/es/deusto/serialization/GenericUserInfo.java](#), 123  
[src/main/java/es/deusto/serialization/LoginAttempt.java](#), 123  
[src/main/java/es/deusto/serialization/OrganizerInfo.java](#), 123  
[src/main/java/es/deusto/serialization/PostInfo.java](#), 123  
[src/main/java/es/deusto/serialization/SignupAttempt.java](#), 124  
[src/main/java/es/deusto/serialization/TopicInfo.java](#), 124  
[src/main/java/es/deusto/serialization/UserInfo.java](#), 124  
[src/main/java/es/deusto/server/dao/DAOFactory.java](#), 124  
[src/main/java/es/deusto/server/dao/EventDAO.java](#), 125  
[src/main/java/es/deusto/server/dao/OrganizerDAO.java](#), 125  
[src/main/java/es/deusto/server/dao/PostDAO.java](#), 125  
[src/main/java/es/deusto/server/dao/TopicDAO.java](#), 125  
[src/main/java/es/deusto/server/dao/UserDAO.java](#), 126  
[src/main/java/es/deusto/server/data/Event.java](#), 126  
[src/main/java/es/deusto/server/data/GenericUser.java](#), 126  
[src/main/java/es/deusto/server/data/Organizer.java](#), 126  
[src/main/java/es/deusto/server/data/Post.java](#), 127  
[src/main/java/es/deusto/server/data/Topic.java](#), 127  
[src/main/java/es/deusto/server/data/User.java](#), 127  
[src/main/java/es/deusto/server/server/AppService.java](#), 127  
[src/main/java/es/deusto/server/server/Server.java](#), 128  
[src/test/java/es/deusto/junit/client/ControllerTest.java](#), 128  
[src/test/java/es/deusto/junit/serialization/SerializationTest.java](#), 128  
[src/test/java/es/deusto/junit/server/ServerTest.java](#), 128  
[startUp](#)  
     [es.deusto.junit.client.ControllerTest](#), 39  
     [es.deusto.junit.serialization.SerializationTest](#), 86  
     [es.deusto.junit.server.ServerTest](#), 96  
[storeEvent](#)  
     [es.deusto.server.dao.EventDAO](#), 52  
[storeOrganizer](#)  
     [es.deusto.server.dao.OrganizerDAO](#), 71  
[storePost](#)  
     [es.deusto.server.dao.PostDAO](#), 80  
[storeTopic](#)  
     [es.deusto.server.dao.TopicDAO](#), 104  
[storeUser](#)  
     [es.deusto.server.dao.UserDAO](#), 111  
[suite](#)  
     [es.deusto.junit.client.ControllerTest](#), 39  
     [es.deusto.junit.serialization.SerializationTest](#), 86  
     [es.deusto.junit.server.ServerTest](#), 96  
[tearDown](#)  
     [es.deusto.junit.server.ServerTest](#), 96  
     [tearDownChild](#)  
     [es.deusto.junit.client.ControllerTest](#), 39  
     [testCreateDeleteEvent](#)  
         [es.deusto.junit.client.ControllerTest](#), 39  
         [es.deusto.junit.server.ServerTest](#), 97  
     [testCreatePost](#)  
         [es.deusto.junit.server.ServerTest](#), 97  
     [testEventInfoFromEvent](#)  
         [es.deusto.junit.serialization.SerializationTest](#), 86  
     [testEventInfoSetters](#)  
         [es.deusto.junit.serialization.SerializationTest](#), 86  
     [testNormalLogin](#)  
         [es.deusto.junit.client.ControllerTest](#), 39  
     [testNormalSignup](#)  
         [es.deusto.junit.server.ServerTest](#), 97  
     [testNormalSignupAndDelete](#)  
         [es.deusto.junit.client.ControllerTest](#), 39  
     [testNormalUpdate](#)  
         [es.deusto.junit.server.ServerTest](#), 97  
     [testOrganizerInfoFromOrganizer](#)  
         [es.deusto.junit.serialization.SerializationTest](#), 86  
     [testOrganizerInfoSetters](#)  
         [es.deusto.junit.serialization.SerializationTest](#), 86  
     [testOrganizerLogin](#)  
         [es.deusto.junit.client.ControllerTest](#), 39  
         [es.deusto.junit.server.ServerTest](#), 97  
     [testOrganizerSignup](#)  
         [es.deusto.junit.server.ServerTest](#), 97  
     [testOrganizerSignupAndDelete](#)  
         [es.deusto.junit.client.ControllerTest](#), 39  
     [testOrganizerUpdate](#)  
         [es.deusto.junit.server.ServerTest](#), 97  
     [testPostInfoFromPost](#)  
         [es.deusto.junit.serialization.SerializationTest](#), 86  
     [testPostInfoSetters](#)  
         [es.deusto.junit.serialization.SerializationTest](#), 86  
     [testSignupAttemptOrganizer](#)  
         [es.deusto.junit.serialization.SerializationTest](#), 87  
     [testSignupAttemptUser](#)  
         [es.deusto.junit.serialization.SerializationTest](#), 87  
     [testUserInfoFromUser](#)  
         [es.deusto.junit.serialization.SerializationTest](#), 87  
     [testUserInfoSetters](#)  
         [es.deusto.junit.serialization.SerializationTest](#), 87  
     [testUserInfoStringConstructor](#)  
         [es.deusto.junit.serialization.SerializationTest](#), 87  
     [testUserUpdate](#)  
         [es.deusto.junit.client.ControllerTest](#), 40  
[Topic](#)  
     [es.deusto.server.data.Topic](#), 102  
[TopicDAO](#)  
     [es.deusto.server.dao.TopicDAO](#), 104  
[TopicInfo](#)  
     [es.deusto.serialization.TopicInfo](#), 105, 106  
[toString](#)  
     [es.deusto.serialization.LoginAttempt](#), 66



- es.deusto.serialization.SignupAttempt, [101](#)
- es.deusto.serialization.TopicInfo, [106](#)
- es.deusto.server.data.Event, [49](#)
- es.deusto.server.data.GenericUser, [61](#)
- es.deusto.server.data.Organizer, [70](#)
- es.deusto.server.data.Topic, [103](#)
- es.deusto.server.data.User, [109](#)
  
- updateEvent
  - es.deusto.server.dao.EventDAO, [52](#)
- updateOrganizer
  - es.deusto.server.dao.OrganizerDAO, [71](#)
- updatePost
  - es.deusto.server.dao.PostDAO, [80](#)
- updateTopic
  - es.deusto.server.dao.TopicDAO, [104](#)
- updateUser
  - es.deusto.server.dao.UserDAO, [111](#)
  - es.deusto.server.server.AppService, [26](#)
- UserDAO
  - es.deusto.server.dao.UserDAO, [110](#)
- UserEventsWindow
  - es.deusto.client.windows.UserEventsWindow, [112](#)
- UserInfo
  - es.deusto.serialization.UserInfo, [113](#)