# ManagementClient

Der ManagementClient baut RMI Connections zum AnalyticsServer und zum BillingServer auf. Wenn dies passiert ist können User diesen ManagementClient mithilfe folgender Kommandos bedienen:

* !login <username> <password>

Loggt einen User mit username und Passwort am BillingServer ein. Nun ist das ausführen von nachfolgenden Secure-Kommandos möglich.

Secure-Kommandos an den BillingServer:

* !steps

Listet alle existierenden „Price Steps“.

* !addStep <startPrice> <endPrice> <fixedPrice> <variablePricePercent>

Fügt einen “Price Step” mit angegebenen Parametern hinzu.

* removeStep <startPrice> <endPrice>

Löscht einen bestehenden „Price Step“

* !bill <userName>

Zeigt die Rechnungen des angegebenen Users an

* !logout

Loggt den user wieder aus

Kommandos an den AnalyticsServer:

* !subscribe <filterRegex>

Meldet den User füe Events mit einem bestimmten Filter an.

* !unsubscribe <subscriptionID>

Meldet den User von einer Anmeldung mit der angegebenen ID ab.

* !auto

Durch diesen Befehl werden alle erhaltenen Events automatisch ausgegeben.

* !print

Durch diesen Befehl werden alle erhaltenen Events gespeichert, jedoch nicht ausgegeben.

* !hide

Durch diesen Befehl werden alle gespeicherten Events ausgegeben.

Die Commands wurden durch das Command-Pattern umgesetzt, darauf ergibt sich ein Interface Command, eine abstrakte Klasse SecureCommand, welche Command implementiert, eine CommandFactory in welcher die jeweiligenCommands, abhängig vom UserInput, erstellt werden sowie alle benötigten Commands. (AddStep, RemoveStep, Steps, Logout,

Testbeschreibung

# AddStepTest

|  |  |  |
| --- | --- | --- |
| **TestCase** | **Input** | **Expected Output** |
| addStepWrongArgumentTest | !addStep muh 1 2 3 | WrongInputException |
| addStepWrongNumberTest | !addStep 1 1 2 | WronNumberOfArgumentsException |

# BillTest

|  |  |  |
| --- | --- | --- |
| **TestCase** | **Input** | **Expected Output** |
| billWrongNumberTest | !bill bob 1 | WronNumberOfArgumentsException |

# CommandFactoryTest

|  |  |  |
| --- | --- | --- |
| **TestCase** | **Input** | **Expected Output** |
| createCommandLoginTest | !login test | instanceOf Login |
| createCommandSecureTest | !steps | CommandIsSecureException |
| createSecureCommandNotFound | !bla | CommandNotFoundException |
| createCommandNotFound | !bla | CommandNotFoundException |

# RemoveStepTest

|  |  |  |
| --- | --- | --- |
| **TestCase** | **Input** | **Expected Output** |
| removeStepWrongInputTest | !removeStep muh 1 | WrongInputException |
| removeStepWrongNumberTest | !removeStep 1 1 2 | WronNumberOfArgumentsException |

# StepsTest

|  |  |  |
| --- | --- | --- |
| **TestCase** | **Input** | **Expected Output** |
| stepsWrongNumberTest | !steps 1 1 | WronNumberOfArgumentsException |

# ManagementClientTest

|  |  |  |
| --- | --- | --- |
| **TestCase** | **Input** | **Expected Output** |
| loginWrongPwTest | !login test muh | “Wrong password!” |
| loginNoBillingTest | Bs.shutdown  !login tes tes | ERROR: Connection to BillingServer lost. You have to login again! |
| printTest | !print | printAutomatic==False |
| autoTest | !auto | printAutomatic==True |
| hideTest | !hide | printAutomatic==False |
| subscribeNoAnalyticeTest | As.shutdown  !subscribe .\* | ERROR: AnalyticsServer not available right now. Retry after starting Analytics |
| subscribeAnalyticsBackTest | As.shutdown  New ManagementClient  New AnalyticsServer  !subscribe .\* | INFO: Analytics seemed to have moved. Looking up |
| unsubscribeAnalyticsBackTest | As.shutdown  New ManagementClient  New AnalyticsServer  !unsubscribe 0 | INFO: Analytics seemed to have moved. Looking up |
| unsubscribeNoAnalyticeTest | As.shutdown  !unsubscribe 0 | ERROR: AnalyticsServer not available right now. Retry after starting Analytics |
| unsubscribeExceptionTest | !subscribe | ERROR: Wrong number of arguments given!\nUsage: !unsubscribe <subscriptionID> |
| subscribeExceptionTest | !unsubscribe | ERROR: Wrong number of arguments given!\nUsage: !subscribe <filterRegex> |
| logoutNoBillingTest | Bs.shutdown  !login test test | ERROR: Connection to BillingServer lost. You have to login again! |
| secureNoBillingTestend | !login test test  Bs.shutdown  !steps | ERROR: Connection to BillingServer lost. You have to login again! |
| endUnsecureTest | !end | Management Client is shutting down! |
| endSecureTest | !login test test  !end | Management Client is shutting down! |
| endSecureNoBillingTest | !login test test  Bs.shutdown  !end | ERROR: Connection to BillingServer lost. You have to login again! |
| nonsenseTest | !muh | ERROR: This Command does not exist!  Allowed commands:  !login <username> <password>  !logout  !steps  !addStep <startPrice> <endPrice> <fixedPrice> <variablePricePercent>  !removeStep <startPrice> <endPrice>  !bill <userName>  !subscribe <filterRegex>  !unsubscribe <subscriptionID>  !print  !auto  !hide |

LoadTester

Der LoadtestingComponent dient dazu, die Robustheit und Skalierbarkeit des AuktionsSystems zu testen.

Es gibt hierzu ein loadtest.properties File, welches die Test-Parameter festlegt.

Bsp:

# TODO: adjust these values

clients = 100

auctionsPerMin = 1

auctionDuration = 2\*60

updateIntervalSec: 20

bidsPerMin = 2

Die Properties Klasse bietet die Funktionalität, ein solches File auszulesen und ebendiese Parameter zu speichern.

Die LoadTest Klasse liest nun diese Parameter aus, und erstellt die gewünschte Menge an Clients.

Von jedem dieser Clients werden mithilfe des TaskExecutors die betreffenden Befhle in den gewünschten Intervallen ausgeführt.

Testbeschreibung

# PropertiesTest

|  |  |  |
| --- | --- | --- |
| **TestCase** | **Input** | **Expected Output** |
| setFromFileErrorTest | Pfad: iwo/lloadtest.properties | FileNotFoundException |
| setAllTest | p.setX.(10)  X = alle jeweiligen attribute | 10  Bei allen gesetzten Attributen |