# File Catalogue

## CurrencyConversionRS

### Source Packages

currencyconversionrs/RSCurrencyConversion

* This class uses hardcoded currency exchange rates stored in a Hashtable. The RESTful service responds only to GET requests and will produce JSON if the path is correct and includes the available currencies. The currency conversion always converts the given currency to GBP and then to a new currency. For example, if a user wanted to exchange USD to EUR, the currency conversion would be USD -> GBP -> EUR and this can be seen in the convert method.

## online-payment-service

### Web Pages

admins/admin

* This is the default admin page after an admin has been authenticated with links to perform all admin functionality on separate pages.

admins/registration

* This is the admin registration page where authenticated admins can register new admins through a form.

admins /viewAllPaymentTransactions

* This web page displays all payment transactions between all users and displays the currency exchange, amount, sender and recipient usernames. This is linked to an adminPaymentTransactionBean which is linked to an EJB which uses a NamedQuery to get all payment data from the database.

admins/viewAllUserAccounts

* This web page displays all user accounts using a datatable linked to userAccountBean. This is linked to an EJB which uses a NamedQuery to get all User Accounts from the database.

users/payment

* This web page includes a form to make a new payment to another user. It uses a backing userPaymentTransactionBean which takes an input, username and amount and uses an EJB TransactionService to make a payment. The JSF backing bean also does validation to ensure that the user has enough funds when sending a payment.

users/paymentTransactions

* This web page displays all sent and received payments for a user using a backing bean userPaymentTransactionBean. This is linked to the EJB TransactionService which calls NamedQueries on the entity to retrieve user payment information.

users/request

* This web page includes a form to make a new request to another user. It uses a backing userRequestBean which takes an input, username and amount and uses an EJB TransactionService to make a request.

users/requests

* This web page displays all requests other than current received pending requests which are displayed on the user web page. The requests are all split into different categories and uses the backing bean userRequestBean. This bean is linked to the EJB TransactionService which calls NamedQueries on the entity to retrieve user request information.

users/user

* This is the default user page after is accessed after user authentication has been performed with links to perform all user functionality on separate web pages. The user page also includes incoming requests which can be accepted and declined to automatically make payments to other users.

error

* This web page is used in the login-config when there is an error when a user does not successfully interact with the Realm.

index

* This is the default web page which lets users register, users and admins to login and allow users and admins to logout.

login

* This is the login page for users and admins which uses a form to login.

registration

* This web page allows users to register. It uses userRegistrationBean which uses the RegistrationService EJB to register new users.

### Source Packages

ejb/converter

* This singleton bean creates a new instance of a client and interacts with the CurrencyConversion RESTful service. It has one method which takes 3 parameters and will build a target path using these parameters to interact with the service. The method will then retrieve the response and return the response as type Double.

ejb/RegistrationService

* This EJB is used to register both users and admins. Users and admins are registered in the same way however users create user accounts which interacts with Converter to get the correct currency value. All objects are then persisted into the database through an EntityManager. Declarative security annotations are used to ensure that each security group can interact with the correct functionality.

ejb/TransactionService

* This EJB manages all functionality regarding user payments and requests. It includes many methods to get information from the database through NamedQueries for both users and admins based on the relevant role. It includes lots of declarative security annotations, splitting user and admin functionality. The EJB also includes methods to make payments, make requests and respond to requests. The method makePayment is a container-based transaction which uses the EntityManager flush method.

ejb/UserService

* This EJB is used to get information about a user. It includes methods to call NamedQueries to retrieve user information from the database.

entity/CurrencyType

* This is an enum for the different currency symbols.

entity/Payment

* This is an entity to model a payment

entity/Request

* This is an entity to model a request

entity/SystemUser

* This is an entity to model a system user

entity/SystemUserGroup

* This is an entity to model a system user group

entity/UserAccount

* This is an entity to model a user account

jsf/AdminPaymentTransactionBean

* This JSF bean uses the TransactionService EJB to get all payments and interact with the NamedQuery to retrieve payment data from the database.

jsf/AdminRegistrationBean

* This JSF bean uses the RegistrationService EJB to register new admins.

jsf/LoginBean

* This JSF bean provides functionality for the user or admin to logout.

jsf/UserAccountBean

* This JSF bean uses the UserService EJB to get information about the user using the SessionContext and interacting with NamedQueries on the UserAccount entity.

jsf/UserPaymentTransactionBean

* This JSF bean uses the TransactionService EJB to make payments and retrieve data about sent and received payments.

jsf/UserRegistrationBean

* This JSF bean uses the RegistrationService EJB to register new users.

jsf/UserRequestBean

* This JSF bean uses the TransactionSerive EJB to make requests, accept requests, decline requests, and retrieve data about requests.