This file is a copy of the very same document, but renamed to reflect a text change.

* “A nut for a jar of tuna” this is an example of a palindrome sentence. A palindrome sentence is those that can be spelled the same way forward an backward. Into the next table on the right side, write down the step-by-step (natural language, NO CODE) to determine if a sentence is a palindrome or not. On the left side list all the java functions you can use to solve this problem.

|  |  |
| --- | --- |
| **Java Functions** | **Step - by - step or algorithm** |
| isPalindrome | 1. Ask for input string to be considered as a Palindrome Text 2. Given String must remove spaces and convert it to lowercase. 3. Also Given String must remove spaces and convert it to lowercase and then reverse it 4. Compare the Result from step 2 with the Result in step 3 and if they are the same then Input String is Palindrome. |

If you want to see a Java implementation refer to the project ***app\_code*** folder and there study the files *PalindromeController, PalindromeServiceImpl, PalindromeControllerTests and PalindromeServiceTests.*

You can point a Chrome Browser to ***http://[server]:[port]/api/v1/palindrome/check?t={text}*** there bBy supplying the {text} you can check if that supplied text is a palindrome one.

* You have a piggy bank, you can only insert coins of the following denomination.

50, 100, 200, 500 & 1000

Keeping in mind the previous description:

* Represent the logic of a piggy bank using java code. means that you can insert but not remove coins.
* You have the option to know how many coins are in the piggy bank.
* You have the option to how many coins are by a specific denomination.

If you want to study the Java Implementation for the **Piggy Bank** Problem use the following files.

* *PiggyBankController, PiggyBankServiceImpl, PiggyBankServiceTests and PiggyBankControllerTests.* These file are the core for the Piggy Bank System implementing the requirements describe above, but it adds a dependency to the Coin Manager (which responsibility is to handle the allowed coins denominations, however, the implementation goes behond and with parameterization via Spring (using application.yml) it allows to registered a list of all coin denominations needed.
* CoinController, CoinServiceImpl, CoinServiceTests and CoinControllerTests are used to hold manage coin denominations, i.e., if a coin denomination is not therein, then **Piggy Bank** cannot allow you to insert the coin.
* Files in the other source folders (packages) are helpers in implementation details.

To see all the system in action, a web REST API was built (given the Controller files mentioned above). Thus in order to use the system, use one of the following **REST API** links (this links are also described in the README.adoc file).

**REST End Points**

To demonstrate the application, there are a few REST points to exercise the options implemented.

Here [server] points to where you have deployed this application, and [port] is the pint where it listens inside the [server].

**Coin Service End Points**

If you want to see the *Coin System* in action use this link.

* ***http://[server]:[port]/api/v1/coin/list*** Shows the valid coin values used for saving coins in the Piggy Bank.

**Pigyy Bank System**

If you want to see the *Piggy Bank System* in action use these links.

* ***http://[server]:[port]/api/v1/piggy-bank/status*** Refers to the way the Piggy Bank internal information is handled.
* ***http://[server]:[port]/api/v1/piggy-bank/insert/{coin}*** Asks the Piggy Bank to insert another coin. If the coin is invalid it will report with a error/msg.
* ***http://[server]:[port]/api/v1/piggy-bank/size*** This tells how many coins (total) there are in the Piggy Bank.
* ***http://[server]:[port]/api/v1/piggy-bank/saved/coins*** This tells how many coins per coin denomination there are in the Piggy Bank.
* ***http://[server]:[port]/api/v1/piggy-bank/remove*** Try to withdraw a coin, but action is not allowed.

**Palindrome System**

If you want to see *Palindrome System* in action use this link.

* ***http://[server]:[port]/api/v1/palindrome/check?t={text}*** By supplying the {text} you can check if that supplied text is a palindrome one.