**Moodle Improvement Project**

**(CMIS Integration Tool Document)**

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Contributions** | **Remarks** |
| 11/10/2019 | Draft 1.0 | Alexis Nelson  Harinder Singh | Document captures the architecture, code structure and code execution details |
| 15/10/2019 | 2.0 | Alexis Nelson  Harinder Singh | Core Data Control Flow sequence added |

Contents

[1 Moodle Improvement Project 4](#_Toc22046965)

[1.1 Architecture 4](#_Toc22046966)

[1.2 Code Structure 5](#_Toc22046967)

[1.3 Code Execution 6](#_Toc22046968)

[1.3.1 Sequence 1 6](#_Toc22046969)

[1.3.2 Continue Sequence 1 7](#_Toc22046970)

[1.3.3 Continue Sequence 2 8](#_Toc22046971)

[1.3.4 Core Data Flow 9](#_Toc22046972)

# ****1 Moodle Improvement Project****

## 1.1 Architecture

The code is based on architectural pattern (MVC framework) that separates an application into three main logical components:

* Model
* View
* Controller

**Important – This code runs as backend process hence it is without “View” components. “View” component is discussed only for the sake of explaining MVC framework.**

**Model**: Model represents shape of the data and business logic. It maintains the data of the application. Model objects retrieve and store model state in a database.

Note - Model is a data and business logic.

**View**: View is a user interface. View display data using model to the user and also enables them to modify the data.

Note- View is a User Interface.

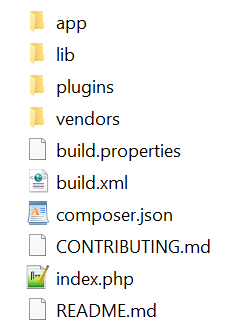
**Controller**: Controller handles the user request. Typically, user interact with View, which in-turn raises appropriate URL request, this request will be handled by a controller. The controller renders the appropriate view with the model data as a response.

Below diagram shows how MVC works:-



## 1.2 Code Structure

Code implementation is using CakePHP (an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [web framework](https://en.wikipedia.org/wiki/Web_framework)). Below diagram shows the home directory structure for the code:-



APP folder – It contains all the files related to Moodle improvement project

LIB folder – It contains CakePHP related files

PLUGINS folder – No files

VENDORS folder - No files

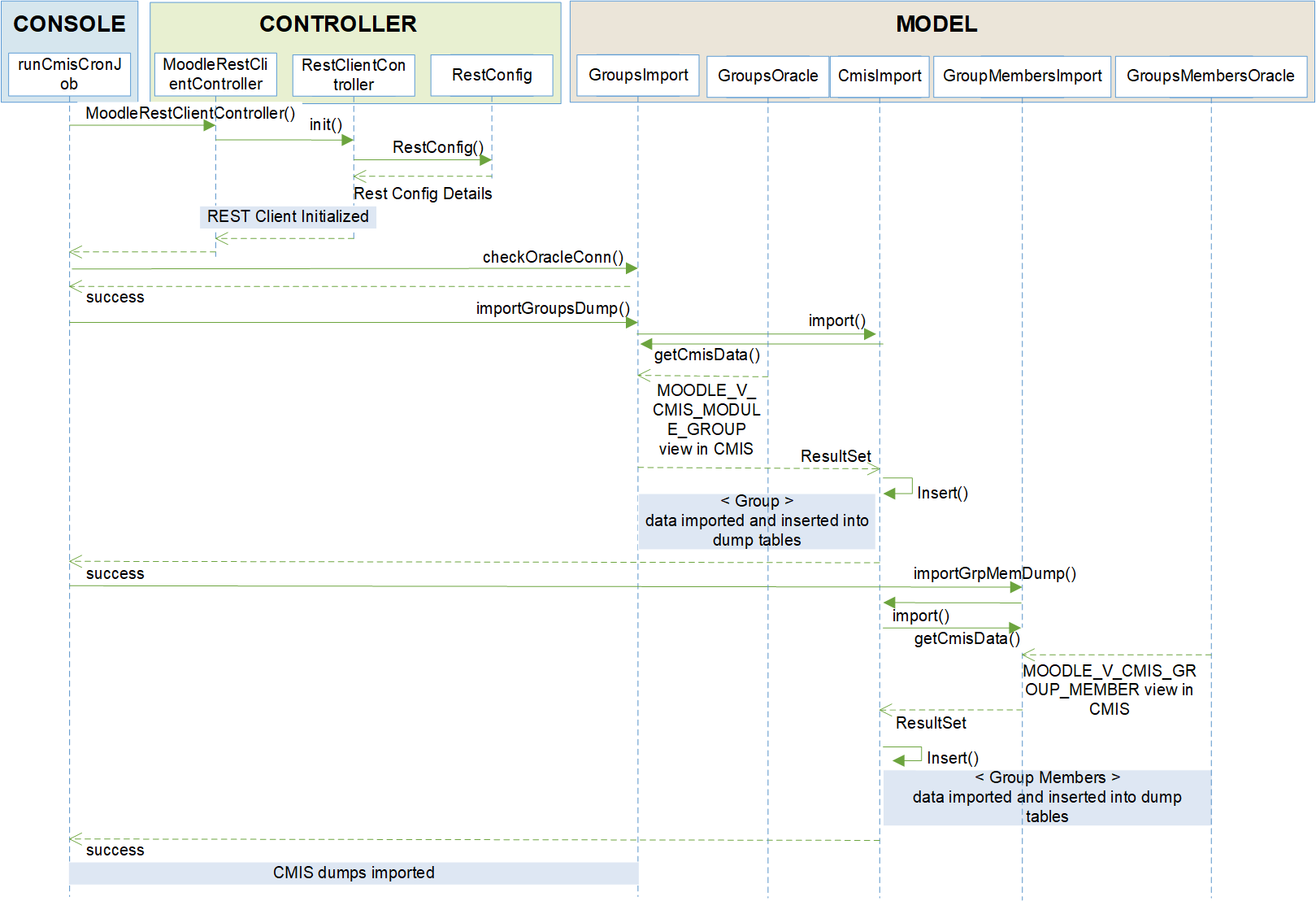
## 1.3 Code Execution

The execution runs through runCmisCronJob function in app\Console\Command \RunCmisShell.php file. Respective controller and models calls are made for achieving the desired functionality.

### 1.3.1 Sequence 1

Below diagram shows execution flow between CONSOLE, CONTROLLER and MODEL functions for

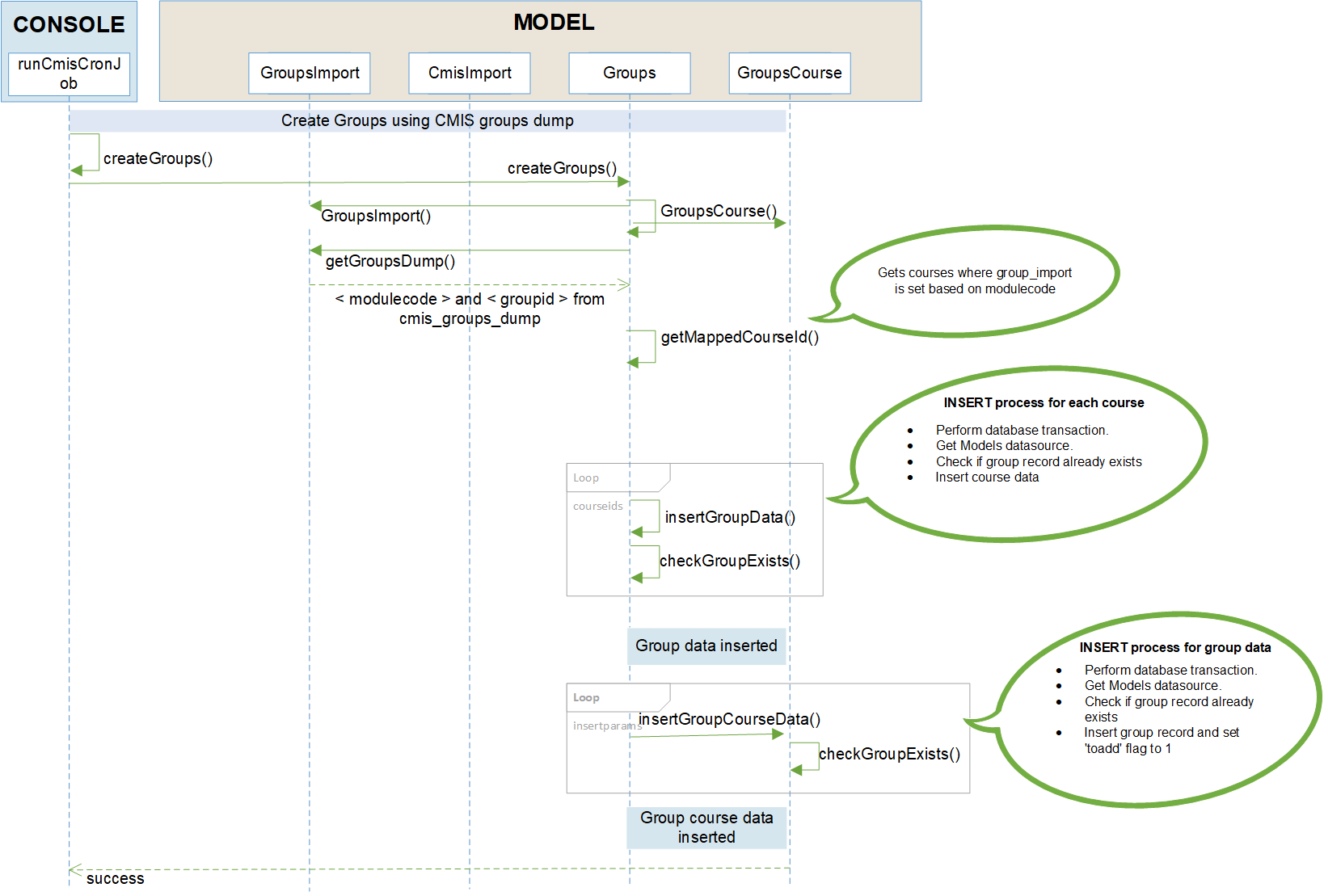
1. REST client initialization
2. Group data import and insertion into dump tables
3. Group members data import and insertion into dump tables
4. CMIS dumps import



### 1.3.2 Continue Sequence 1

Below diagram shows execution flow between CONSOLE and MODEL functions for

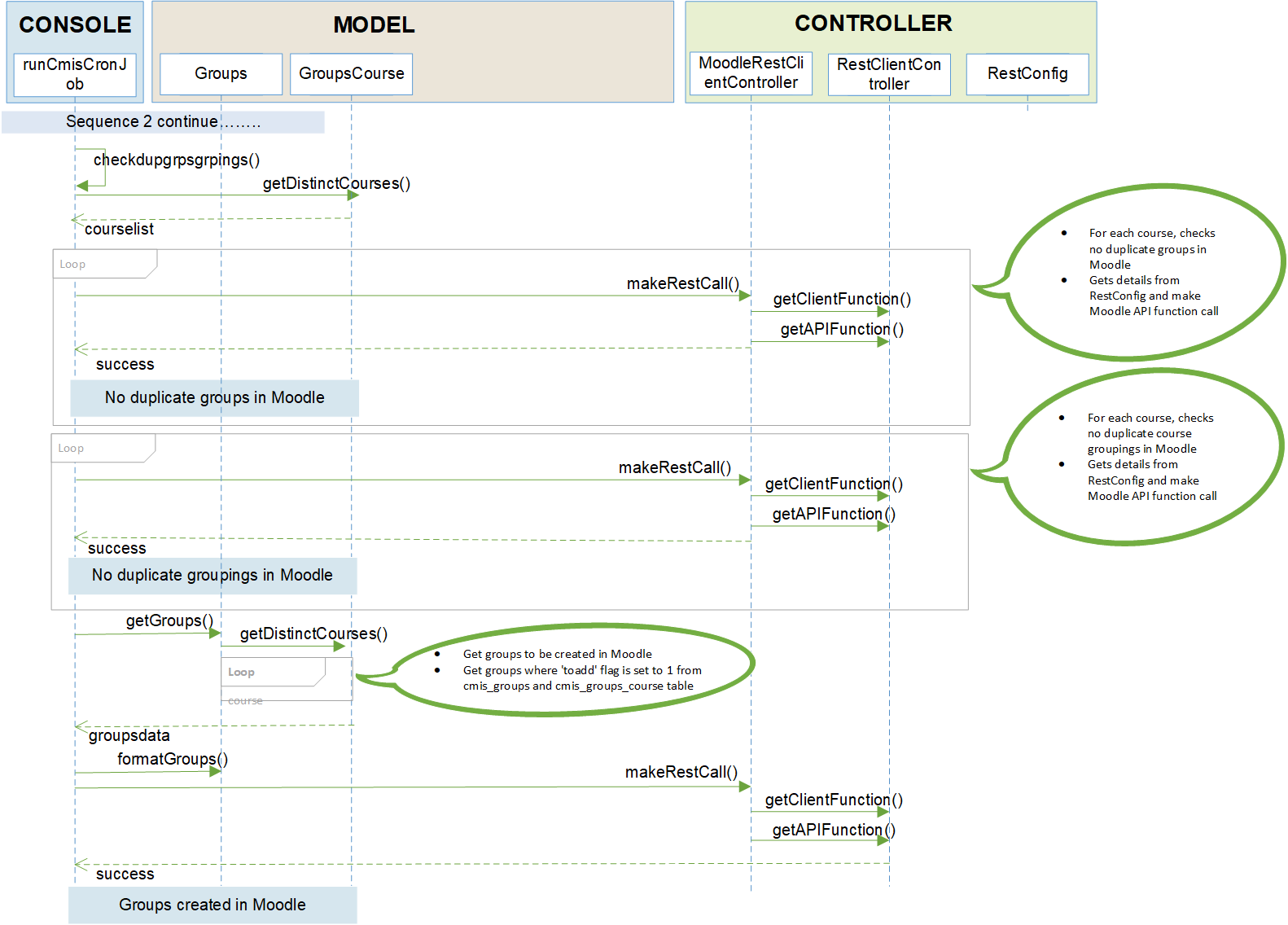
1. Group data insertion
2. Group course data insertion



### 1.3.3 Continue Sequence 2

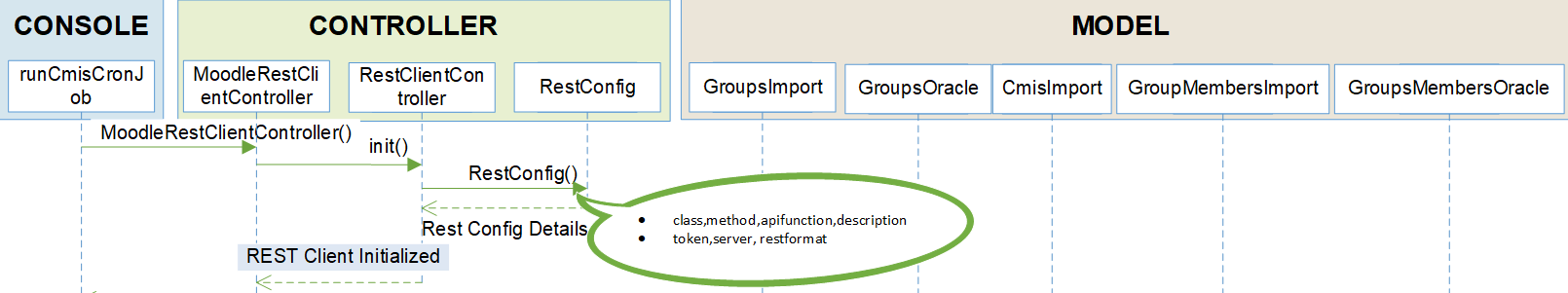
Below diagram shows execution flow between CONSOLE, MODEL and CONTROLLER functions for

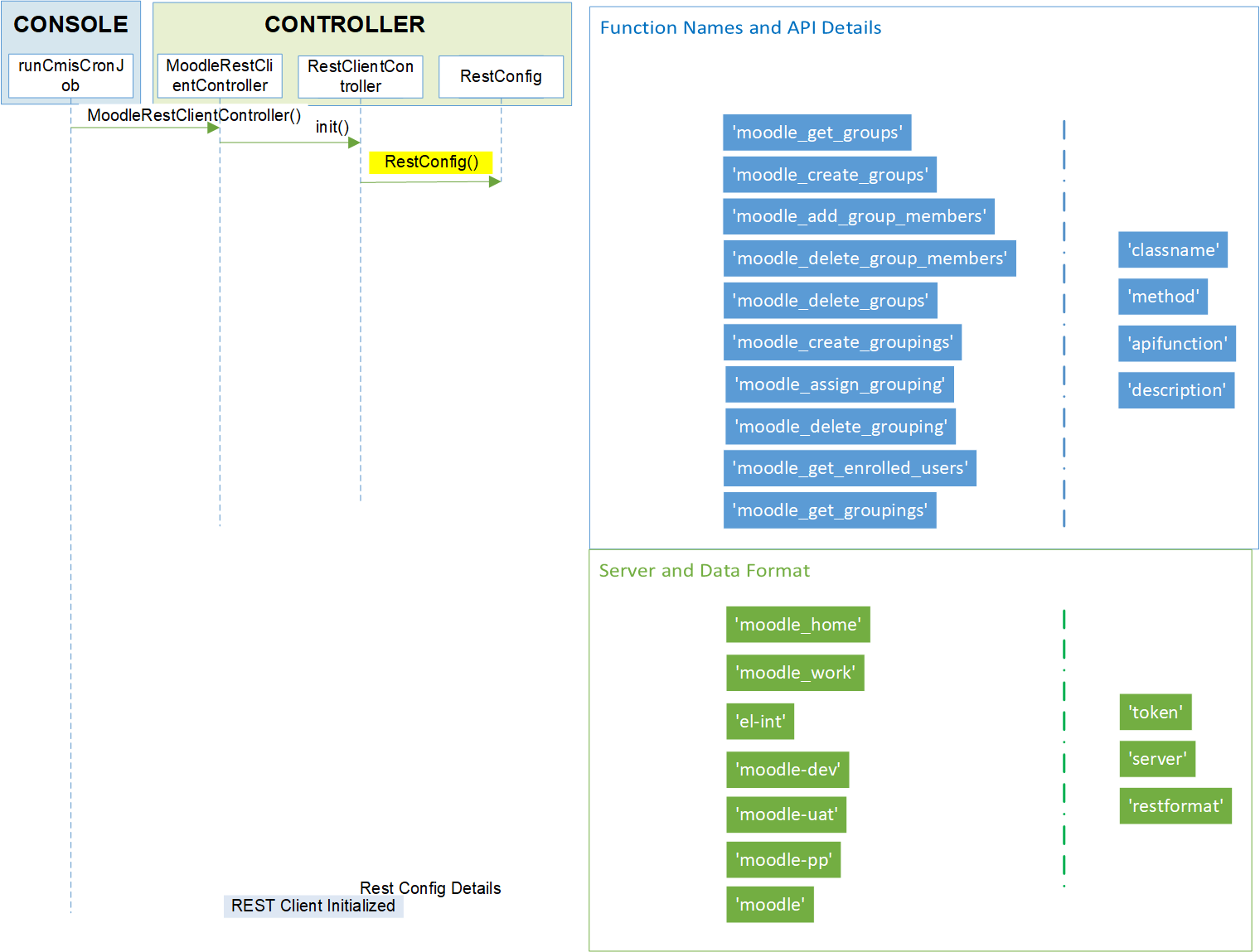
1. Group duplication checking in Moodle
2. Groupings duplication checking in Moodle
3. Groups creation in Moodle



### 1.3.4 Core Data Flow

**REST Config**





**REST API CALL**

