

---

# **The Comodojo Framework Documentation**

*Release 1.0.0*

**Marco Giovinazzi**

**Apr 06, 2019**



---

## Contents

---

<b>1</b>	<b>Project Status</b>	<b>3</b>
<b>2</b>	<b>Install Comodojo</b>	<b>5</b>
2.1	Requirements . . . . .	5
2.2	Installing via composer . . . . .	5
<b>3</b>	<b>Applications</b>	<b>7</b>
3.1	The application schema . . . . .	7
3.2	The client side . . . . .	7
3.3	The server side . . . . .	7
3.4	Packaging applications . . . . .	7
<b>4</b>	<b>Users and Roles</b>	<b>11</b>
<b>5</b>	<b>Authentication</b>	<b>13</b>
<b>6</b>	<b>Themes</b>	<b>15</b>
6.1	Theme package schema . . . . .	15
<b>7</b>	<b>Appendix A: the Comodojo Installer</b>	<b>17</b>
<b>8</b>	<b>Indices and tables</b>	<b>19</b>





Comodojo is a PHP/js framework that provides a complete environment to realize complex and easy-to-maintain web applications.

Instead of offer a common foundation classes to build a single web app, Comodojo aims to be a pre-built environment where:

- a couple of logical related function (or even a single one) can be realized by an application;
- an application publishes routes both on a server and client side, defining state-aware, linkable locations;
- applications can (and generally should) be deployed, updated, removed using composer; in addition, multiple applications can be packaged under the same composer package;
- internal role-mapper provides the connection between users and applications, exposing to user only the routes it should reach.

<p><b>Warning:</b> Code is under active development, there is no stable release and it may change without notice at any time.</p>
---

---

**Note:** If you're interested in participating development, please send a message to [marco.giovinazzi@comodojo.org](mailto:marco.giovinazzi@comodojo.org).

---

Contents:



# CHAPTER 1

---

## Project Status

---

The framework is still in a pre-alpha stage and there is no planned release date yet.

Code will be published on [packagist](#) as soon as an alpha release is available.

Many of the foundation libraries are already stable and released under MIT license (refer to [comodojo website](#) for a complete list) as well as the two microframework [dispatcher](#) and [extender](#).





Comodojo can be installed via [composer](#), using [comodojo/comodojo](#) project's package.

### 2.1 Requirements

To work properly, comodojo requires an apache webserver (nginx will also be supported ) with PHP >=5.6.0.

### 2.2 Installing via composer

**Warning:** This procedure is not complete, unstable, untested. In other words, there are no chances it will work.

A new comodojo/comodojo project can be created with:

```
composer create-project comodojo/comodojo myinstallationpath
```

The web root of your server should point to myinstallationpath/public.

More details will come in near future.



(Section yet to be written)

### 3.1 The application schema

(Section yet to be written)

### 3.2 The client side

(Section yet to be written)

### 3.3 The server side

(Section yet to be written)

### 3.4 Packaging applications

Application can be packed into bundles using special composer package type *comodojo-app*.

```
{
  "name": "my/app",
  "description": "My first comodojo app",
  "type": "comodojo-app",
  "extra": {
    "comodojo-app-register": {
      "helloworld": {
        "description": "My first comodojo app",
```

(continues on next page)

```
        "assets": "assets"
    }
}
"comodojo-configuration-register": {
    "app-helloworld-default": {
        "value": 'Comodojo',
        "constant": false,
        "type": "STRING",
        "validate": ""
    }
}
"comodojo-rpc-register": {
    "myrpc.helloworld": {
        "callback": "\\My\\Rpc",
        "method": "helloworld",
        "description": "Rpc Helloworld",
        "signatures": [
            {
                "returnType": "string",
                "parameters": {
                    "name": {
                        "type": "string",
                        "optional": true
                    }
                }
            }
        ]
    }
}
"comodojo-service-route": {
    "helloworld": {
        "type": "route",
        "class": "\\My\\Helloworld",
        "parameters": {
            "cache": false
        }
    }
}
"comodojo-task-register": {
    "HelloWorld": {
        "class": "\\My\\Tasks\\HelloWorldTask",
        "description": "Greetings from comodojo"
    }
},
"comodojo-command-register": {
    "helloworld": {
        "class": "My\\App\\Command\\Helloworld",
        "description": "Greetings from comodojo",
        "aliases": ["hw"],
        "options": {},
        "arguments": {
            "to": {
                "choices": {},
                "multiple": false,
                "optional": true,
                "description": "hello to..."
            }
        }
    }
}
```

(continues on next page)

(continued from previous page)

```
        }
    }
},
"autoload": {
    "psr-4": {
        "My\\App\\": "src"
    }
}
}
```



## CHAPTER 4

---

### Users and Roles

---

(Page yet to be written)





## CHAPTER 5

---

### Authentication

---

(Page yet to be written)



(Page yet to be written)

### 6.1 Theme package schema

```
{
  "name": "my/theme",
  "description": "My first comodojo theme",
  "type": "comodojo-components",
  "extra": {
    "comodojo-theme-register": {
      "mytheme": {
        "description": "My first comodojo theme",
        "assets": "assets"
      }
    }
  }
}
```



## CHAPTER 7

---

### Appendix A: the Comodojo Installer

---

(Page yet to be written)



## CHAPTER 8

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`