

Model

User
<div>- dataTable: 'user'</div> <div>- fname: string</div> <div>- lname: string</div> <div>- email: string</div> <div>- password: string</div> <div>- user_uuid: string</div>
<div>+ getUserId(): object</div> <div>+ getEmail(object): string</div> <div>+ authenticate(string, string, array): bool</div> <div>+ registerUser(string, string, string, string, string):bool</div> <div>- confirmEmail(string, string): bool</div> <div>- confirmPassword(string, string): bool</div> <div>- saltPassword(string): string</div> <div># createUserSession(object): void</div> <div>+public destroyUserSession(): void</div>

Post
<div>- dataTable: 'posts'</div> <div>- body: string</div> <div>- title: string</div> <div>- post_uuid: string</div>
<div>- getUserId(): string</div> <div>+ getPosts(): array</div> <div>+ getPostByPostUuid(string): object</div> <div>+ createPost(string, string): bool</div> <div>+ editPost(string, string, string): bool</div> <div>+ deletePost(string): bool</div> <div>- checkFields(string, string): bool</div>

View

Includes
<div><div><div><div>&lt;&lt;Interface&gt;&gt;</div><div>Header</div><div><div>- &lt;head&gt;</div><div>+ bootstrap</div><div>+ fontawesome:</div></div></div></div><div><div><div>&lt;&lt;Interface&gt;&gt;</div><div>Footer</div><div><div>- &lt;script&gt;:jquery</div><div>- &lt;script&gt;:cloudfare</div><div>- &lt;script&gt;:bootstrap</div></div></div></div><div><div><div>&lt;&lt;Interface&gt;&gt;</div><div>Navbar</div><div><div>+ &lt;navbar&gt;:view</div><div><div>- isLoggedIn():bool</div><div>- getField(string):string</div><div>+ logoutPath(string):controller</div><div>+ loginPath(string):controller</div><div>+ registerPath(string):controller</div></div></div></div></div></div>
Pages

<b>&lt;&lt;Interface&gt;&gt; Not_Found</b>
+ loginPath(string):controller

**Posts**

<b>&lt;&lt;Interface&gt;&gt; Index</b>
+ fname:string + lname:string + dateCreated:date + title:string + title:body
+ foreach(array):array + postsEdit(string):controller + postsDelete(string):controller

<b>&lt;&lt;Interface&gt;&gt; Edit</b>
+ postsEdited(array):string + title(array):string + body(array):string

<b>&lt;&lt;Interface&gt;&gt; Delete_Error</b>
+ postsHome(string):controller

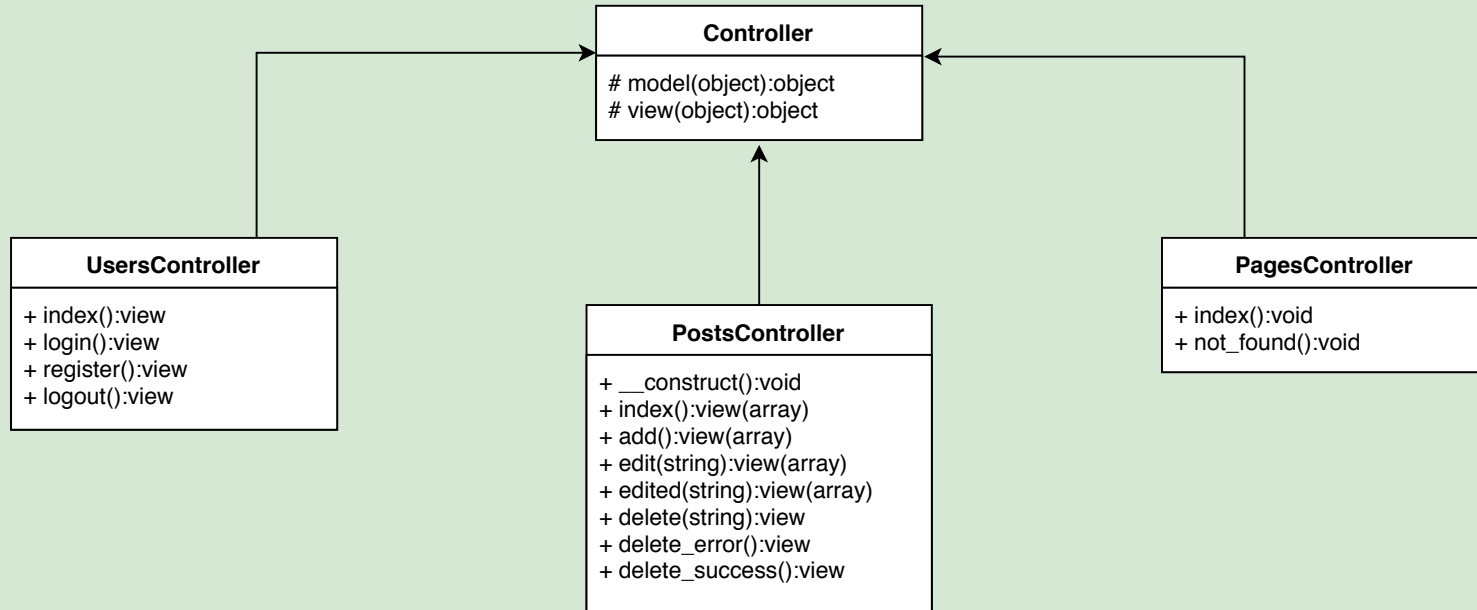
<b>&lt;&lt;Interface&gt;&gt; Delete_Success</b>
+ postsHome(string):controller

**Users**

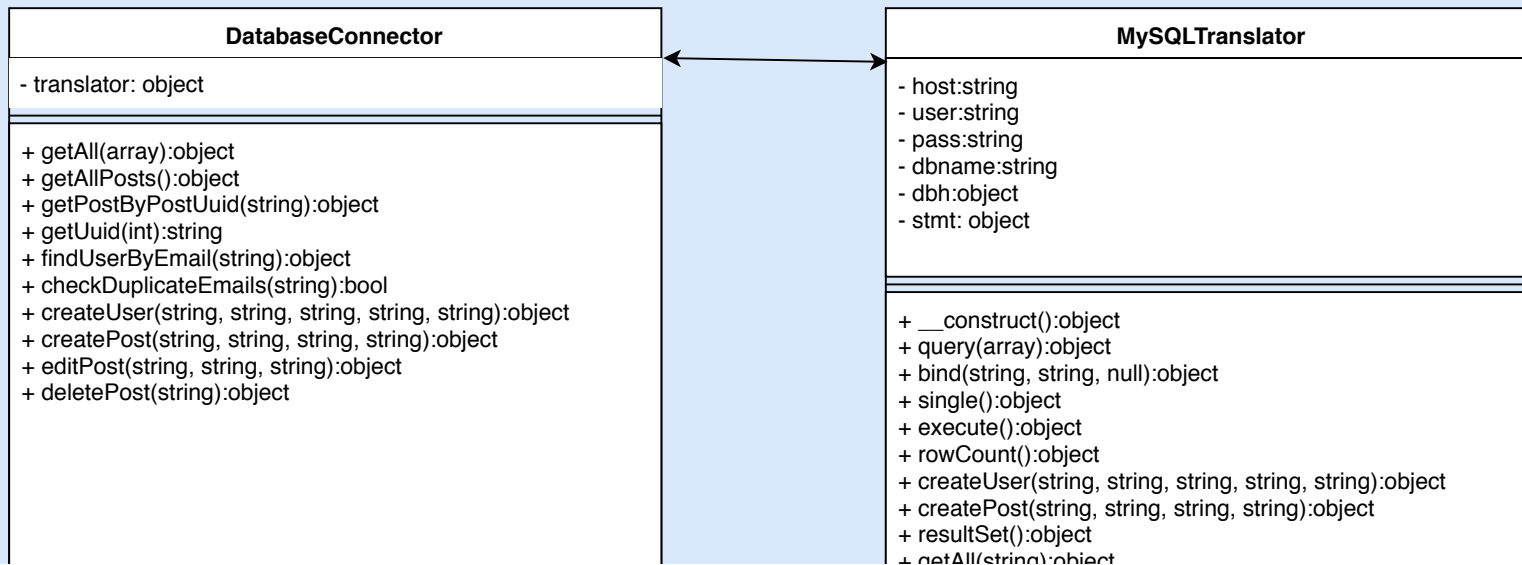
<b>&lt;&lt;Interface&gt;&gt; Index</b>
+ form:array + email:string + password:string
+ button(array):controller + registerPath(string):controller

<b>&lt;&lt;Interface&gt;&gt; Register</b>
+ form:array + fname:string + lname:string + email:string + confirmEmail:string + password:string + confirmPassword:string
+ button(array):controller + registerPath(string):controller

## Controller



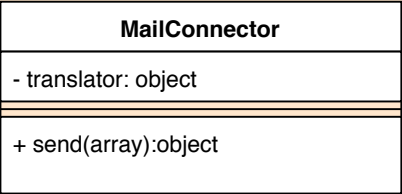
## Data



**API**

```
classDiagram
    class MailConnector {
        - translator: object
        + send(array):object
    }
    class SendGridTranslator {
        + send(array):object
    }
    MailConnector <--> SendGridTranslator
```

The diagram illustrates the API structure with two classes: **MailConnector** and **SendGridTranslator**. **MailConnector** has a private attribute `- translator: object` and a public method `+ send(array):object`. **SendGridTranslator** has a public method `+ send(array):object`. A bidirectional association arrow connects the two classes, indicating they interact with each other.



**Libraries**

```
classDiagram
    class Redirect {
        +to(string):view
    }
    class Session {
        +getField(object):object
        +isPost():string
        +sanitizePost():object
        +fieldIsSet(object):object
        +isLoggedIn():bool
    }
    class SprayItCore {
        -currentController:string
        -currentMethod:string
        -params:array
        +__construct():object
        +getURL():string
    }
```

The diagram illustrates three libraries used in the application:

- Redirect**: Contains a method `+ to(string):view`.
- Session**: Contains methods `+ getField(object):object`, `+ isPost():string`, `+ sanitizePost():object`, `+ fieldIsSet(object):object`, and `+ isLoggedIn():bool`.
- SprayItCore**: Contains attributes `- currentController:string`, `- currentMethod:string`, and `- params:array`, along with methods `+ __construct():object` and `+ getURL():string`.

