Tài liệu thiết kế

Mục lục

URL 3

Bài viết 3

Hành động (action) 3

Cơ sở dữ liệu 3

Tags 4

actionButton & actionLink 4

articleButton & articleLink 4

userLink 4

form 5

parse 5

Functions 6

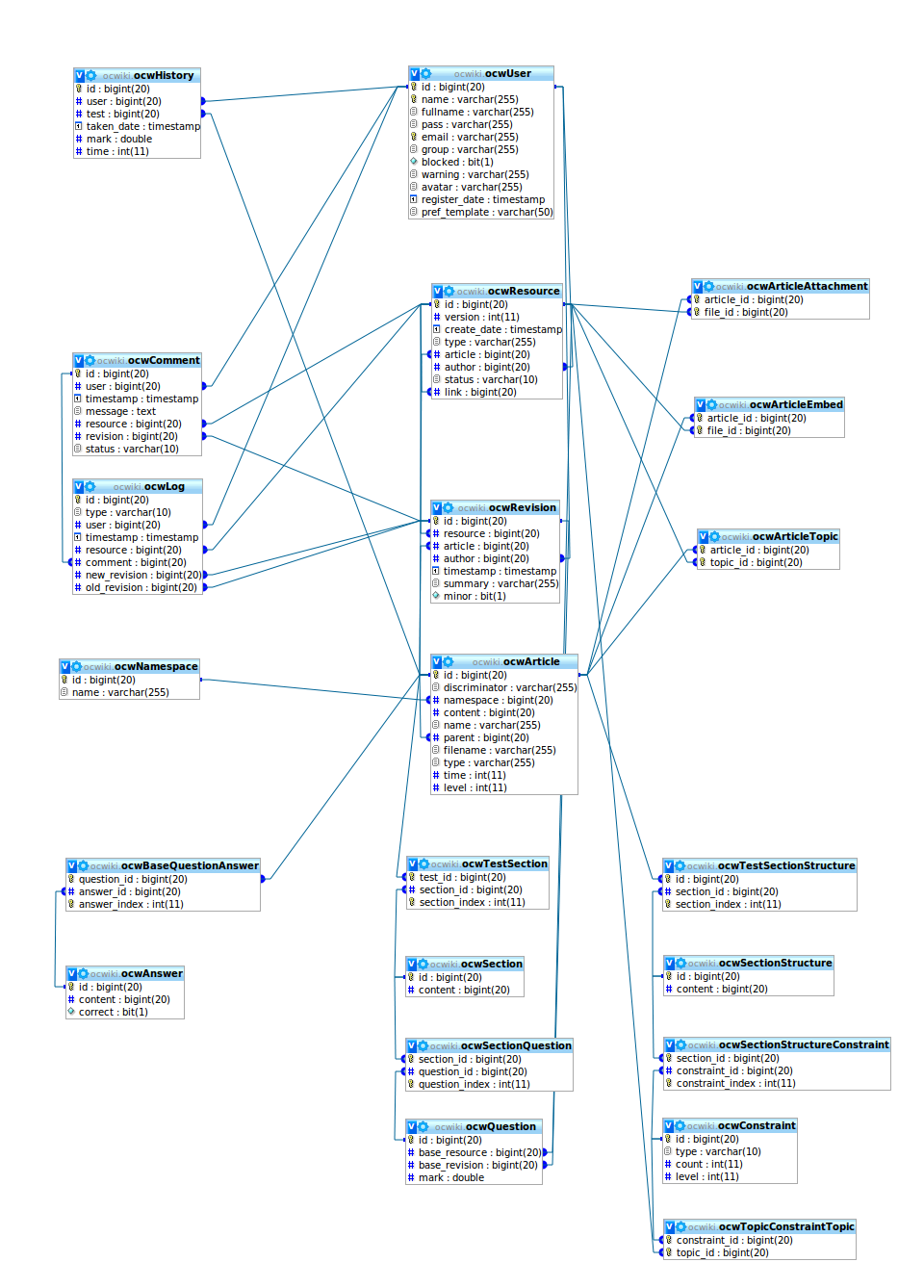
* 1. Cơ chế thực hiện hành động
  2. URL
     1. Bài viết

${homeDir}/article/${id}

* + 1. Hành động (action)

${homeDir}/action/${code}

* 1. Cơ sở dữ liệu



* 1. Tags
     1. actionButton & actionLink

Tạo liên kết đến hành động, dùng thẻ này để code rõ ràng và dễ thay đổi hơn.

Cú pháp:

<ocw:action(Button|Link) name=... [id=...] [class=...] [confirm=...]>

[<ocw:param name=... value=...>]

content

</ocw:action(Button|Link)>

Trong đó confirm là một lời gọi hàm JavaScript trẻ về true/false.

Ví dụ:

*<ocw:actionLink name="article.changelog">*

*<ocw:param name="article" value="${action.resource}" />*

*Nhật kí*

*</ocw:actionLink>*

* + 1. articleButton & articleLink

<ocw:article(Button|Link) resource=resource\_object>

</ocw:article(Button|Link)>

Trong đó resource\_object là một đối tượng kiểu oop.data.Resource.

Ví dụ:

*<ocw:articleButton resource="${action.resource}">*

*Quay về câu hỏi*

*</ocw:articleButton>*

* + 1. userLink

*<ocw:userLink user=user\_object>*

*content*

*</ocw:userLink>*

*user\_object* là đối tượng kiểu oop.data.User.

* + 1. form

Sinh ra một thẻ form chứa sẵn editToken bên trong.

<ocw:form action=... [id=...] [class=...] [method=...] >

content

</ocw:form>

* + 1. parse

Chuyển đổi văn bản từ định dạng nội bộ sang HTML, khi trình bày bất kì bài viết nào cần bọc phần văn bản trong thẻ này. Cú pháp:

<ocw:parse resource=... [content=...]>

[...]

</ocw:parse>

Trong đó resource là đối tượng lớp oop.data.Resource, trong văn bản có thể có các nội dung động lấy thông tin từ đối tượng này. Thuộc tính content là nội dung cần chuyển đổi, thuộc kiểu String hoặc oop.data.Text. Thuộc tính content là tuỳ chọn, thay vào đó có thể đặt văn bản vào thân thẻ.

* 1. Functions