Frontend Editing

To get Frontend Editing (FE) up and running after installing the package, you need to:

1. Add a property of the data type "Frontend editable" (which is installed with the package) to the content types that should be editable with FE. You can add it to a master content type if you want all child content types to be frontend editable.
2. Add `@Umbracian.FrontendEditing.Helper.Init()` to the bottom of your `<head>` section in the templates for the same content types, or simply to the master layout. This loads the FE scripts and style sheets, but don't worry; it will do absolutely nothing and have no impact on your page rendering time, if there's no FE session active.

To start an FE session, browse to the page you want to edit. Now append "/edit" to the URL and hit enter. This will invoke the alternative template "Edit", which is installed with the package. You can tweak the "Edit" template to your heart's content, but by default it simply starts the FE session and redirects the browser back to the page it came from. FE should now be active and show a login icon in the upper right corner of the browser or, if you're already logged into the backoffice, the FE controls.

FE requires jQuery loaded on the page to work. However, if you're fond of loading your scripts in the bottom of the page, FE will still work, even if the FE scripts and style sheets are loaded in the `<head>` section. If you're not using jQuery on your site, you can conditionally include it for FE usage like this:

```
@if (Umbracian.FrontendEditing.Helper.HasSession()) {
  <script src="//code.jquery.com/jquery-2.1.1.min.js"></script>
}
```

Edit markers

You can add edit markers to the individual page properties. When the user clicks an edit marker, the Umbraco UI will automatically open the tab on which corresponding page property resides and scroll the property into view.

To use edit markers, you must first add the following using statement to your template:

```
@using Umbracian.FrontendEditing;
```

You insert edit markers by surrounding your page properties with `Html.BeginFrontendEditable()` – like this:

```
@using(Html.BeginFrontendEditable("title", "Edit page title")) {
  <h2>@pageTitle</h2>
}
```

The method takes the following arguments:

1. The property alias (e.g. "title")
2. A help text that will be displayed when hovering the edit marker (e.g. "Edit page title"). This argument is optional.

You can also add edit markers conditionally, which might come in handy if you're adding edit markers to a common layout template or partial view. In this case you'll use `Html.BeginConditionalFrontendEditable()` and specify the condition as the first argument:

```
  @using(Html.BeginConditionalFrontendEditable(Model.Content.DocumentTypeAlias ==
"umbHomePage", "aboutText", "Edit about text")) {
    <p>@Umbraco.Field("aboutText", recursive: true)</p>
  }
```

Localization

FE supports localization, so you can translate the UI to your needs. After you've called `@Umbracian.FrontendEditing.Helper.Init()`, you need to hook into the global `ufeLocalize` variable and define a `GetText` method like this:

```
  @if (Umbracian.FrontendEditing.Helper.HasSession()) {
    <script>
      ufeLocalize = {};
      ufeLocalize.GetText = function (key) {
        // translate key to localized text here.
        // return null if there's no translation (use default text)
        return "your-translation-of-key";
      }
    </script>
  }
```

Currently supported localization keys (subject to change) are:

- For the **FE controls**:
  - login.control.title
  - edit.control.title
  - create.control.title
  - unpublish.control.title
  - delete.control.title
  - logout.control.title
- For the **login dialog**:
  - login.dialog.text (supports HTML in localization)
  - login.dialog.username.placeholder
  - login.dialog.password.placeholder
  - login.dialog.button.ok
  - login.dialog.button.cancel
- For the **create dialog**:
  - create.dialog.text (supports HTML in localization)
  - create.dialog.button.ok
  - create.dialog.button.cancel
- For the **delete dialog**:
  - delete.dialog.text (supports HTML in localization)
  - delete.dialog.button.ok
  - delete.dialog.button.cancel
- For the **unpublish dialog**:

    - unpublish.dialog.text (supports HTML in localization)
    - unpublish.dialog.button.ok
    - unpublish.dialog.button.cancel
- For the **edit markers**:
    - editable.helpText.[property alias] for the edit marker help text, e.g. *editable.helpText.title* for the property with alias *title*.