Проектирование

Эта глава посвящена то, что везде и во всем должен быть порядок. Так и в рабочем столе и на кухне и тем более в созданном для конечного пользователя приложении. А если не будет порядка, то и при создании мы получим «хаос!» и неразбериху, которые не позволят нам закончить приложение:

- написали несколько страниц кода и поняли, что упустили в середине какой-то функционал…
- создали страницу с логикой и страница долго грузится…
- и так далее…

Все это и не только, станет большим разочарованием и в большенстве заставит нас отложить разработку, а потом и вообще забросить проект.

Мы должны точно знать, что будет находиться между начальной и конечной точкой создаваемого приложения. То-есть:

- index.php - начальная точка входа в приложение.
- «что-то просходит», к примеру подключается база данных, подключаются стили проекта, формируются таблицы, окрывается форма регистрации и т.д.
- admin_panel.php - конечная точка приложения.

И вот в этом «что-то просходит» будет хаос и путаница и разочарование от написания проекта, если не составить заранее план. Нам неоходимо прописать предполагаемый план план взаимодействия приложения и пользователя, например:

- Пользователь заходит на главную страницу впервые, программа проверит:
  1. если приложение не установленно, то отправит пользователя в панель установки приложеня.
  2. если приложение установленно, то приложение проверит регистрацию пользователя.
  3. если пользователь не зарегистрирован, перенаправит на регистрацию.
  4. если пользователь зарегистрирован, перенаправит на авторизацию.
  5. если авторизация не прошла успешно, заблокирует на время этого пользователя.
  6. если авторизация прошла успешно, откроет панель администратора.

Пока вроде бы легко и просто, но посмотрим на пункт 1.

- если приложение не установленно, то пользователь перенаправляется в панель установки приложеня, а это страница выбора языка и часовой зоны и продолжаем…
  - если пользователь не выбрал язык, то
    1. приложение по умолчанию создаст файл конфигурации и запишет в него язык браузера или первый язык в списке приложения
  - если пользователь не выбрал часовую зону, то
    1. приложение определит время сервера и запишет его в файл конфигурации…

Легко? Вроде бы легко, но мы уже не учли, что после выбора языка приложение должно переключить нашу страницу в выбранный язык, а какие языки у приложения будут? И пользователю после выбора нужно подтвердить кнопкой «Продолжить», а иначе приложение, установит язик по умолчанию, не дождавшись выбора пользователя…

Лугко? Вроды бы «ДА», но как-то не то, что мы ожидали и хочется пропустить этот пунк, но увы, если мы напишем приложение, согласно данной инструкции, то получим приложение, которое уже кто-то написал для учебной цели этой инструкции, и уже есть готовая версия этого приложения. Вы сможите загрузить, как демонстрационную, так и рабочую версию для некоммерческого использования. Прочитав руководство вы сможите модифицировать код под себя и т.д., это позволяет лицензия. Но если вы решите написать свое, то без проекта вам не обойтись, а он в свою очередь облегчит вам работу по написаю кода. Готовы потрать время на проект? Если «Да», продолжаем:

Начало проекта

Запишем основные действия пользователя будущего приложения:

- Пользователь заходит на главную страницу впервые, программа проверит:
    1. если приложение не установленно, то отправит пользователя в панель установки приложеня.
    2. если приложение установленно, то приложение проверит регистрацию пользователя.
    3. если пользователь не зарегистрирован, перенаправит на регистрацию.
    4. если пользователь зарегистрирован, перенаправит на авторизацию.
    5. если авторизация не прошла успешно, заблокирует на время этого пользователя.
    6. если авторизация прошла успешно, откроет панель администратора.
    7. если открыта панель администратора, но не установлена конфигурация (видеохостинг, блог или социальная сеть), то приложение придложит сделать выбор и установить.
    8. если открыта панель администратора и установлена конфигурация (видеохостинг), то приложение придложит сделать выбор между закрытой или открытой.
        1. если открытая, то можно ли регистрироваться и т.д.. (Видеохостинг описан в руководстве по написанию UCMS Video, что является продолжение данного руководства). А в этом, мы остановимся на выборе конфигурации, окне настройки, установке дополнительных модулей (загрузка файлов, почта, редактор текста, двуфакторная авторизация и т.д.), добавление шаблонов, языков и прочее, а пока…

В наш маленький проект внесем дополнения:

- Пользователь заходит на главную страницу **index.php** впервые, программа проверит:
    1. если приложение не установленно (файлы **config.ini**, **version.txt** не существуют), то отправит пользователя в панель установки приложеня (файл **install.php**) в директорию **/install**.
    2. панель установки приложеня (файл **install.php**) предложит выбрать язык (английский/русский)
    3. по умолчанию предложен английский (файл **english.php**) в дирректории **lang** установщика
    4. при выборе русский (файл **russian.php**) в дирректории **lang** установщика **/install** переключит установщик на русский язык и будет ожидать выбора часового пояса (форма выбора **timezone** -12, 0, +12)
    5. после выбора станет активной кнопка «Продолжить», при нажатии на которую,

данные lang_select и timezone_select будут записаны в файл конфигурации **config.ini**
6. пользователь будет перенаправлен к окну проверки совместимости сервера (файл **check_server.php**)

Пока остановимся. Откуда названия файлов и директорий (папок)? Придумываем имена файлам сами, чтобы нам проще было понять, например:

- config - в переводе с английского конфигурация
- install - установка
- lang - язык
- check_server - проверить сервер
- timezone - часовой пояс и т.д.

Таким образом, мы всегда будем понимать, что директория **/install** - это папка установки, а переменная lang_select - выбор языка, стало быть **lang_select_russian** - выбор языка русский.

Подумаем еще немного… А где будет располагаться наше приложение? В корне сайте **./** или в директории **/ucms**! Мы не знаем что хочет пользователь, т.е. сделаем в директории **/ucms** без привязки к директории, а за это будет отвечать файл, где будут постоянные данные - константы (стало быть **cons.php**). Т.к. **cons.php** и **config.ini** являются конфигурационными, то целесообразно использовать отдельную директорию **/config**

Продолжим наш план-проект…

- Пользователь заходит в директорию **/ucms** на главную страницу файл **index.php** впервые, программа проверит:
  1. если приложение не установленно (файлы **config.ini**, **version.txt** не существуют), то отправит пользователя в панель установки приложеня (файл **install.php**) в директорию **/install**.
  2. панель установки приложения (файл **install.php**) предложит выбрать язык (английский/русский)
  3. по умолчанию предложен английский (файл **english.php**) в дирректории **lang** установщика
  4. при выборе русский (файл **russian.php**) в дирректории **lang** установщика **/install** переключит установщик на русский язык и будет ожидать выбора часового пояса (форма выбора **timezone** -12, 0, +12)
  5. после выбора станет активной кнопка «Продолжить», при нажатии на которую, данные lang_select и timezone_select будут записаны в файл конфигурации **config.ini** в дирректории **/config**
  6. пользователь будет перенаправлен к окну проверки совместимости сервера (файл **check_server.php**), который запросит наличие файла **version.txt** и

1. если приложение установленно, то приложение проверит регистрацию пользователя.
2. если пользователь не зарегистрирован, перенаправит на регистрацию.
3. если пользователь зарегистрирован, перенаправит на авторизацию.
4. если авторизация не прошла успешно, заблокирует на время этого пользователя.
5. если авторизация прошла успешно, откроет панель администратора.

From:
https://wwoss.ru/ - **worldwide open-source software**

Permanent link:
https://wwoss.ru/doku.php?id=software:development:demo:cms:ucms:planning&rev=1746637701

Last update: **2025/05/07 20:08**