

ВЕБ-СЛУЖБА

E-WEBINAR.RU

Русская документация программиста

Оглавление

Общие положения	3
Описание	3
Общие указания	3
Требования	3
Разработка	4
Подготовка веб приложения	4
Контроллер.....	5
<i>Инициализация</i>	<i>5</i>
<i>Регистрация.....</i>	<i>5</i>
Все данные полученные от службы являются объектами и имеют свойство ->status, отображающее статус обращения.....	5
<i>Авторизация</i>	<i>5</i>
<i>Идентификатор пользователя.....</i>	<i>6</i>
Объекты wObject.....	6
<i>Создание объекта.....</i>	<i>6</i>
<i>Загрузка коллекции объектов.....</i>	<i>6</i>
<i>Получение значений:</i>	<i>7</i>
<i>Изменение:</i>	<i>7</i>
<i>Сохранение:.....</i>	<i>7</i>
Инвайты.....	7
<i>Создание инвайта</i>	<i>7</i>
<i>Регистрация по инвайту.....</i>	<i>8</i>
Использование шаблонов.....	8
<i>Загрузка шаблона из файла.....</i>	<i>8</i>
<i>Поля шаблона.....</i>	<i>8</i>
<i>Связка данных с полями шаблона.....</i>	<i>9</i>
<i>Рендеринг шаблонов</i>	<i>9</i>
<i>Вложенность шаблонов.....</i>	<i>9</i>
Отладка	9



Общие положения

Описание

Данная веб-служба предназначена для интеграции функционала для проведения веб-конференций(далее вебинар) в различные веб-проекты партнеров.

Для интеграции используется набор (далее программный модуль) php-скриптов, позволяющий в короткие сроки при минимальных трудозатратах внедрить необходимый функционал в проект.

Обмен данными базируется на протоколе SOAP. При этом не используется схема WSDL. Нет необходимости разрабатывать клиент для обмена данными, т.к. он уже написан и готов к использованию.

Общие указания

Каждый партнер, использующий данный сервис должен получить кодовое слово `company_codename` для работы с сервисом, которое определяет отдельное пространство для работы и участвует в авторизации. Кодовое слово относится к служебному виду информации и не должно распространяться без согласия владельца сервиса.

Требования

Модуль не требует подключения каких-либо расширений php и настройки сервера. Для работы по умолчанию необходима поддержка сессий. Версия php должна быть не ниже 5.2



Разработка

Подготовка веб приложения

Для работы с модулем необходимо внести дополнения в те скрипты, которым будет необходим доступ к веб-службе. В случае, когда используется CMS или фреймворк дополнительные строки можно внести в инициализирующий скрипт (`index.php`).

1. Необходимо инициализировать константу WEBINAR_PATH

```
define('WEBINAR_PATH', $_SERVER['DOCUMENT_ROOT'].'/webinar/');
```

2. Подключить класс главного контроллера

```
require_once WEBINAR_PATH.'core.php';
```

3. Определить место хранения учетных данных пользователей (по умолчанию это `$_SESSION['webinar_user']`) внести необходимые изменения в файл `config.php`.

```
'storage' => array (
    'adapter' => 'session',
    'key'      => 'webinar_user'
),
```

4. Заполнить следующие поля массива значениями текущего пользователя:

```
if (!isset($_SESSION['webinar_user'])) {
    $_SESSION['webinar_user'] = $_SESSION['webinar_user'];
    unset($_SESSION['webinar_user']['id']);
    'username' => 'Vano',
    'email' => 'ivanec@mail.ru',
    'surname' => 'Ivanov',
    'name' => 'Ivan',
    'middlename' => 'Ivanovich',
    'password' => 'password' // оставлять пустым, если не нужен
    функционал входа через форму
}
```

Внимание! В массиве не должен присутствовать ключ `id`.

5. Данные пользователя должны быть в переменной сессии до первого обращения к контроллеру модуля.



Контроллер

Практически вся работа с модулем сводится к работе с главным контроллером. Все классы подключаются по мере необходимости.

Инициализация

```
$controller = Webinar::instance();
```

При создании объекта-контроллера происходит автоматическая авторизация. Если пользователь не авторизирован нужно его зарегистрировать.

Регистрация

```
$query = $webinar->register($_SESSION['webinar_user']);
if ($query->status == 'REGISTERED')
    echo 'New user was succesfully registered';
```

Все данные полученные от службы являются объектами и имеют свойство `->status`, отображающее статус обращения.

Статусами любого обращения могут быть:

`DENIED` — пользователь не авторизирован

`ERROR` — ошибка базы данных, неверно переданные параметры, ошибка модели данных

Подробное сообщение об ошибке содержится в поле `->message` возвращаемого объекта.

Так объект ответа при регистрации может иметь следующие статусы:

`REFUSED` — незарегистрированный `company_codename`

`REGISTERED` — пользователь успешно зарегистрирован

`ALREADY_REGISTERED` — пользователь уже зарегистрирован

В случае успешной регистрации пользователь становится авторизованным автоматически.

Авторизация

Выполнять авторизацию нет необходимости, при инициализации контроллера она происходит автоматически. Но если вы хотите использовать форму для входа вам понадобится следующая функция:

```
$controller->login('email','password');
```



Идентификатор пользователя

В целях безопасности модуль не работает с идентификаторами пользователя напрямую, отсылая каждый раз службе идентификатор сессии, полученный от службы. Для подключения к вебинару необходим идентификатор пользователя. Получить его можно используя следующий метод:

```
$user_id = Webinar::instance()->getUserId();
```

Объекты wObject

Работа с вебинарами как с сущностью сводится к работе с объектами. Создать объект вебинара, готовый к работе можно вызовом метода главного контроллера.

Создание объекта

```
$webinar = Webinar::instance()->factory();
```

Если передать параметр `$id` контроллер загрузит объект вебинара с сервиса.

```
$webinar = Webinar::instance()->factory($id);
```

Возможные статусы при загрузке вебинара по идентификатору:

NOT_FOUND — вебинар не был найден по идентификатору
 REFUSED — вебинар не принадлежит партнеру
 LOADED — вебинар успешно загружен

Загрузка коллекции объектов

```
$controller = Webinar::instance();
$all_webinar = $controller->all();           // получить все вебинары по company_codename
$my_webinars = $controller->own();           // получить вебинары пользователя
$sub_webinarss = $controller->subscribed(); // получить вебинары по приглашению
```

Возможные статусы загрузки коллекции:

EMPTY — вебинары не найдены
 LOADED — вебинары загружены успешно

Если объекты вебинаров были загружены успешно, массив этих объектов можно получить следующим образом:

```
$array_of_wObjects = $all_webinars->objects
```



Получение значений:

```
$data = $webinar->values();
```

Изменение:

```
$webinar->values($_POST);
```

Или

```
$webinar->title = $data;
```

Сохранение:

```
$result = $webinar->save();
```

Внимание! Для создания и сохранения используется один и тот же метод. Если объект вебинара не содержит **id**, будет создан новый вебинар.

Инвайты

Инвайты позволяют приглашать других пользователей на вебинар (звать друзей). Приглашения отсылаются на почту и должны содержать инвайт код, по которому возможна регистрация на нашем сервисе. Инвайт код полностью пригоден для передачи его в качестве GET параметра.

Создание инвайта

```
$invite = $webinar->invite(array('email'=>'',  
                                'surname'=>'', 'name'=>'', 'middlename'=>'', 'password' => '' ));
```

Если вы не собираетесь использовать авторизацию с использованием формы ввода для приглашенных пользователей, не передавайте параметр **password**.

Возможные статусы при создании инвайта:

INVITED — пользователь успешно приглашен

ALREADY_INVITED — пользователь уже получал инвайт на данный вебинар

Получить код можно из результатов запроса:

```
$code = $invite->code
```



Регистрация по инвайту

Если есть необходимость организовать участие в вебинарах пользователей, незарегистрированных на Вашем сайте, вы можете зарегистрировать их в нашем сервисе по коду инвайта.

```
$reg_by_invite = Webinar::instance()->register($code);
```

Возможные статусы регистрации по инвайту:

FAULT — инвайт не был найден по коду

ALREADY_REGISTERED — пользователь уже зарегистрирован в базе сервиса

REGISTERED — пользователь успешно зарегистрирован

Если регистрация прошла успешно, или пользователь уже был зарегистрирован в базе сервиса, происходит автоматическая авторизация.

При этом в ответе будут содержать и следующие данные:

```
$reg_by_invite->webinar_id; // получить идентификатор вебинара
```

Таким образом вы можете отклонировать страницы с необходимым функционалом и дать к ним доступ для незарегистрированных на вашем сайте пользователей. Зарегистрированные пользователи будут авторизоваться по данным из вашей базы данных через сессию, а приглашенные со стороны пользователи по инвайт кодам.

Внимание! Вы можете не использовать Инвайты и организовать **всю** логику в своем приложении. При этом `$controller->subscribed()` будет возвращать статус **EMPTY**.

Использование шаблонов

Для облегчения внедрения в папке **templates** модуль поставляется с шаблонами, код которых вы можете исправить под себя.

Загрузка шаблона из файла

```
$template = Webinar::instance()->template('webinars/all');
```

Поля шаблона

Поля в шаблоне - именованные места для подстановки значений, которые обозначаются двойными квадратными скобками `[[]]` по умолчанию:

```
echo '<div><a href="[[create]]">Создать вебинар</a></div>';
```



Поменять скобки на другие обозначения можно следующим образом:

```
$template->casing('{','}');
```

Связка данных с полями шаблона

Для получения массива полей и их значений для связывания используется метод

```
$vars = $template->vars();
```

Связка значения с полем шаблона:

```
$template->myvar = 'value';
```

Для связки полей шаблона со значениями ассоциативного массива используется следующий метод:

```
$template->bind(array('field' => 'value'));
```

Вы также можете связать значения на этапе инициализации:

```
$template = Webinar::instance()->template('webinar/update', $_POST);
```

Внимание! Значения передаются не по ссылке, внешнее изменение значений после связки не изменит массив `$template->vars()`

Рендеринг шаблонов

```
$template->engine();
```

А также сразу после инициализации:

```
Webinar::instance()->template('webinar/info', webinar->values())->engine();
```

Вложенность шаблонов

Вебинары могут быть вложены друг в друга.

```
$main_template = Webinar::instance()->template('main');
$nested_template = Webinar::instance()->template('other');
$main_template->other_template = $nested_template->parse();
$main_template->engine();
```

Отладка

Для отладки работы и просмотра возвращаемых сервисом данных необходимо включить режим отладки в конфигурационном файле. После этого все директивы

```
webinar::debug($your_var, 'Пояснения к выводу');
```

Будут обрабатываться контроллером и выводить подробную информацию в удобном виде.

