

rCalendar

rCalendar – JavaScript плагин для бронирований столов.

Необходимые компоненты:

- jQuery версии 3 и выше;
- twitter bootstrap 3 и выше.

Инициализация плагина

Инициализация плагина происходит при указании соответствующего класса r-calendar в блоке div и вызова плагина в JavaScript коде.

Пример:

```
<script> $(".r-calendar").rCalendar();</script>  
<div class="r-calendar"></div>
```

Параметры при инициализации

language – параметр выбора языка: ru, en. По умолчанию: ru;

startDay – дата с которой необходимо начинать строить календарь. Передается в формате объекта JS *new Date()*. По умолчанию: текущая дата.

view – режим отображения календаря: months, weeks, days. По умолчанию: months;

selectTable – список столов для бронирования. Передается в виде массива;

selectCustomer – список клиентов для бронирования. Передается в виде массива;

handlersScript – путь к файлу для обработки событий;

arrayDataEvents – JSON-массив с данными для обработки данных.

Примеры инициализации

1) Инициализация в режиме просмотра «Дни»:

```
$(".r-calendar").rCalendar({ "view" : "days" });
```

2) Инициализация с выбранной датой (02.04.2020):

```
var date = new Date(2020, 04, 02);  
$(".r-calendar").rCalendar({ "startDay" : date });
```

3) Инициализация с выбранным языком:

```
$(".r-calendar").rCalendar({ "language" : "en" });
```

4) Инициализация с массивом столов для бронирования:

```
var selectTable = { "1" : "1-й стол", "2" : "2-й стол" };  
$(".r-calendar").rCalendar({ "selectTable" : selectTable });
```

```
var selectTable = ["1-й стол", "2-й стол"];  
$(".r-calendar").rCalendar({ "selectTable" : selectTable });
```

5) Инициализация с массивом клиентов для бронирования:

```
var selectCustomer = {"1" : "1-й клиент", "2" : "2-й клиент"};  
$(".r-calendar").rCalendar({ "selectCustomer" : selectTable });
```

```
var selectCustomer = ["1-й клиент", "2-й клиент"];  
$(".r-calendar").rCalendar({ "selectCustomer" : selectTable });
```

6) Инициализация с указанием скрипта для обработки событий:

```
$(".r-calendar").rCalendar({ "handlersScript" : "handlers.php" });
```

7) Инициализация с начальным массивом данных:

```
var data = [{ "id" : "1", "startDate" : "17.04.2020", "startDateHour" : "2", "startDateMinute" : "00",  
: "endDate" : "17.04.2020", "endDateHour" : "4", "endDateMinute" : "55",  
"nameReservation" : "Дружеская встреча", "tableReservation" : "2-й стол",  
"customerReservation" : "3-й клиент", "commentReservation" : "Комментарий"},  
{ "id" : "2", "startDate" : "13.04.2020", "startDateHour" : "13", "startDateMinute" : "00",  
"endDate" : "13.04.2020", "endDateHour" : "16", "endDateMinute" : "00", "nameReservation" :  
"Дружеская встреча", "numberTableReservation" : "2-й стол", "customerReservation" : "3-й клиент",  
"commentReservation" : "Комментарий"}];  
$(".r-calendar").rCalendar({ "arrayDataEvents" : data });
```

Передача параметров для обработки событий

С помощью метода **AJAX** осуществляется передача данных к обработчику событий, путь к которому задается параметром **handlersScript**. Доступно несколько методов:

select – получения списка бронирований;

insert – вставка нового бронирования;

update – обновление существующего бронирования;

delete – удаление бронирования.

Все параметры передаются обработчику с использованием метода **POST** в формате **JSON-массива**.

Метод **SELECT**: { "action" : "select", "startDate" : dd1, "endDate" : dd2 }

Метод **INSERT**: { "action" : "insert", "startDate" : dd1, "startDateHour" : x1, "startDateMinute" : x2, "endDate" : x3, "endDateHour" : x4, "startDateMinute" : x5, "nameReservation" : x6, "tableReservation" : x7, "customerReservation" : x8, "commentReservation" : x9, "id" : "-1" }

Метод **UPDATE**: { "action" : "update", "startDate" : dd1, "startDateHour" : x1, "startDateMinute" : x2, "endDate" : x3, "endDateHour" : x4, "startDateMinute" : x5, "nameReservation" : x6, "tableReservation" : x7, "customerReservation" : x8, "commentReservation" : x9, "id" : "1" }

Метод **DELETE**: { "action" : "delete", "id" : "1" }

От сервера в зависимости от операции ожидается в формате JSON массива:

Метод **SELECT**: { "OK", *arrayDataEvents* }

Метод **INSERT**: { "OK", "id_answer" }

Метод **UPDATE**: { "OK" }

Метод **DELETE**: { "OK" }