1. Тестовое задание для фронтенд-разработчика

Необходимо написать **плагин для jquery** для выбора региона и вывода списка городов выбранного региона. Плагин должен применяться к div элементу и генерировать в нем input[type=text] для ввода названия региона. При вводе названия региона, должен отправляться запрос на API, который возвращает список найденных регионов. Этот список необходимо отображать в виде ссылок на странице. При клике по ссылке названия региона, должен отправляться запрос на другой API интерфейс, который возвращает список городов для выбранного региона. Полученный список городов необходимо выводить на странице в виде обычного текста. При изменении значения в поле «название региона» список найденных регионов должен обновляться, а список городов очищаться. Список регионов и городов очищается, если не введено название региона.

При начальной инициализации плагина может быть передан ID региона (см. пример использования плагина), для которого нужно подгрузить по API список городов и вывести их на странице в соответствующем блоке. Если при инициализации не передан ID региона, то список городов будет пустым. После инициализации должна работать логика описанная выше.

Пример отображения данных

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | Инпут для ввода региона | | Список найденных регионов в виде ссылок | | Список найденных городов выбранного региона |

Дополнительный функционал

Плагин может быть вызван с рядом настроек (см. «Варианты использования»). После вызова плагина, должна быть возможность установить настройки.

|  |
| --- |
| 1. $('.selector').geoSelector('set','minInputChar'); |

Также, должна быть возможность получить настройки объекта, к котрому применялся плагин.

|  |
| --- |
| 1. var gs=$('.selector').geoSelector('get','minInputChar'); |

Во время запроса данных по регионам, при вводе пользователем, функционал не должен допускать большого количества одновременных запросов. Если пользователь быстро печатает – зачем нам сразу «валить» все на сервер, создавая запросы к серверу на каждую букву.

1. Варианты использования
2. Вариант 1: *Обычное применение к объектам*

|  |
| --- |
| 1. $('.selector').geoSelector(); |

1. Вариант 2: *Применить, по умочанию выбрать регион с id = 56, загрузит города для него*

|  |
| --- |
| 1. $('.selector').geoSelector(56); |

1. Вариант 3: В*ызов с настройками*

|  |
| --- |
| 1. $('.selector').geoSelector({ 2. regionID: [integer], // Регион по умолчанию 3. minInputChar: [integer], // Минимальноче количество символов 4. // в имени региона, после которого будет    1. // производиться запрос на поиск, по умолчанию = 1 5. bindEvent: [string], // Событие, на которое "повешан" поиск, 6. // по умолчанию "keyup" 7. events: { // Набор событий, для работы с данными и отлова ситуаций 8. onGsData: function(data) {}, // Происходит при получении 9. // списка регионов с сервера, данные в data 10. onGsError: function(text) {}, // Происходит при ошибке, текст ошибки в text 11. onGsCityStart: function() {}, // Происходит сразу перед отправкой 12. // запроса на поиск городов 13. onGsCityData: function(data) {}, // Происходит при получении списка 14. // городов с сервера, данные в data 15. } 16. }); |

1. API интерфейсы

Поиск регионов

|  |  |
| --- | --- |
| URL | http://evildevel.com/Test/Region |
| TYPE | GET |
| **Принимаемые данные** | |
| name | Строка, название региона по маске «%X%». |
| **Возвращаемые значения** | |
| JSON объект со списком регионов.  Пример:  [  [1, «Курганская область»],  [5, «Курская область»]  … другие регионы подходящие под условие  ] | |

Поиск городов

|  |  |
| --- | --- |
| URL | http://evildevel.com/Test/City |
| TYPE | GET |
| **Принимаемые данные** | |
| region | Целое число, ID региона. |
| **Возвращаемые значения** | |
| JSON объект со списком городов.  Пример:  [  [14, «Москва»],  [18, «Балашиха»],  [23, «Коломна»]  … другие города региона  ] | |

\* \* \* \* \*

Пишите мне в скайп «evildevel.com» если у вас есть вопросы по заданию.