# Interface Definition- 接口定义

## Standard Web API - 基本Web API接口 -

1. RequestResult SelectAll() - Collection query for objects themselves单表集合查询
2. RequestResult SelectAllWithChild () – Collection query for objects with aggerated children objects. 集合附加子对象查询
3. RequestResult SelectById(int Id) – Single object query by id 单表查询
4. RequestResult SelectByIdWithChild(int id, bool isAggregatedChildren) – Single object and its aggerated children objects query by id 附加子对象查询
5. RequestResult SelectBy(RequestData requestData) Generic collection query by multiple conditions 单表集合查询
6. RequestResult Insert(Entity entity) Create a new object单表新增
7. RequestResult Update(Entity entity) Update an object单表修改
8. RequestResult Delete (int id) Delete an object单表删除
9. RequestResult SelectPagerByColumns(RequestData requestData) Generic collection query by multiple conditions with paging capability 分页单表集合查询

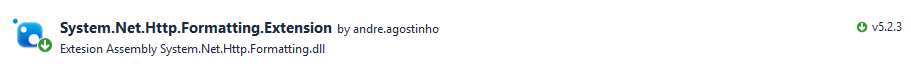
## Version Control Web API - 版本控制WebApi接口

1. RequestResult SelectBySpecifyDate (int Id , DateTime timeStamp) Query effective versioned object 时间戳对应版本并且附加子对象的查询
2. RequestResult SelectAllByDateTime (DateTime timeStamp) Query effective versioned object collection 时间戳对应版本集合查询
3. RequestResult SelectAllByDateTimeWithChild (DateTime timeStamp) Query effective versioned object collection 时间戳对应版本集合并且附加子对象查询

# 后台调用Api

注意：使用下面的方式调用Api需要安装NuGet包：

Attention: To work with Web API, following package from NuGet is needed in caller’s project.



## SelectAll

注意：下文中的ControllerName写法：比如BlendChemicalApiController在URL中需要写成BlendChemicalApi

Attention: ControllerName naming convention following instructions: e.g. BlendChemicalApiController needs to be written as BlendChemicalApi in URL.

* Call API in C# - 调用方式

HttpClient myHttpClient = new HttpClient();

string url = "http://ipAddress:port/ControllerName/";

myHttpClient.BaseAddress = new Uri(url);

HttpResponseMessage response = myHttpClient.GetAsync("SelectAll").Result;

* Get return - 获取调用结果：

RequestResult result = response.Content.ReadAsAsync<RequestResult>().Result;

List<T> entities = JsonConvert.DeserializeObject<List<T>>(result.Data.ToString());

## SelectAllWithChild

* Call API in C# - 调用方式

HttpClient myHttpClient = new HttpClient();

string url = "http://ipAddress:port/ControllerName/";

myHttpClient.BaseAddress = new Uri(url);

HttpResponseMessage response= myHttpClient.GetAsync("SelectAllWithChild").Result;

* Get return - 获取调用结果：

RequestResult result = response.Content.ReadAsAsync<RequestResult>().Result;

List<T> entities = JsonConvert.DeserializeObject<List<T>>(result.Data.ToString());

## SelectById

* Call API in C# - 调用方式

HttpClient myHttpClient = new HttpClient();

string url = "http://ipAddress:port/ControllerName/";

myHttpClient.BaseAddress = new Uri(url);

HttpResponseMessage response = myHttpClient. GetAsync("SelectById?id="+ id).Result;

* Get return - 获取调用结果：

RequestResult result = response.Content.ReadAsAsync<RequestResult>().Result;

T entity = JsonConvert.DeserializeObject<T>(result.Data.ToString());

## SelectByIdWithChild

* Call API in C# - 调用方式

HttpClient myHttpClient = new HttpClient();

string url = "http://ipAddress:port/ControllerName/";

myHttpClient.BaseAddress = new Uri(url);

HttpResponseMessage response = myHttpClient. GetAsync("SelectByIdWithChild?id="+ id+"&isAggregatedChildren=true").Result;

* Get return - 获取调用结果：

RequestResult result = response.Content.ReadAsAsync<RequestResult>().Result;

T entity = JsonConvert.DeserializeObject<T>(result.Data.ToString());

## SelectBy

* Call API in C# - 调用方式

HttpClient myHttpClient = new HttpClient();

string url = "http://ipAddress:port/ControllerName/";

myHttpClient.BaseAddress = new Uri(url);

List<string> columnNames = new List<string>()

{

"Name"

};

T entity = new T()

{

Name = "name"

};

Dictionary<string, string> requestData = (new RequestData<T> {Entity = entity, Columns = columnNames }).Serialize();

FormUrlEncodedContent content = new FormUrlEncodedContent(requestData);

HttpResponseMessage response = myHttpClient.PostAsync("SelectBy ",content).Result;

* Get return - 获取调用结果：

RequestResult result = response.Content.ReadAsAsync<RequestResult>().Result;

List<T> entities = JsonConvert.DeserializeObject<List<T>>(result.Data.ToString());

## Insert

* Call API in C# - 调用方式

HttpClient myHttpClient = new HttpClient();

string url = "http://ipAddress:port/ControllerName/";

myHttpClient.BaseAddress = new Uri(url);

T entity = new T ();

HttpResponseMessage response = myHttpClient.PostAsJsonAsync("Insert", entity).Result;

* Get return - 获取调用结果：

RequestResult result = response.Content.ReadAsAsync<RequestResult>().Result;

T entity = JsonConvert.DeserializeObject<T>(result.Data.ToString());

## Update

* Call API in C# - 调用方式

HttpClient myHttpClient = new HttpClient();

string url = "http://ipAddress:port/ControllerName/";

myHttpClient.BaseAddress = new Uri(url);

T currentEntity = new T ();

HttpResponseMessage response = myHttpClient.PostAsJsonAsync("Update",currentEntity).Result;

* Get return - 获取调用结果：

RequestResult result = response.Content.ReadAsAsync<RequestResult>().Result;

T entity = JsonConvert.DeserializeObject<T>(result.Data.ToString());

## Delete

* Call API in C# - 调用方式

HttpClient myHttpClient = new HttpClient();

string url = "http://ipAddress:port/ControllerName/";

myHttpClient.BaseAddress = new Uri(url);

HttpResponseMessage response = myHttpClient.GetAsync("Delete?id="+ id).Result;

* Get return - 获取调用结果：

RequestResult result = response.Content.ReadAsAsync<RequestResult>().Result;

T entity = JsonConvert.DeserializeObject<T>(result.Data.ToString());

## SelectPagerByColumns

* Call API in C# - 调用方式

HttpClient myHttpClient = new HttpClient();

string url = "http://ipAddress:port/ControllerName/";

myHttpClient.BaseAddress = new Uri(url);

Pager pager = new Pager()

{

PageIndex = 1,

PageSize = 1

};

List<string> columnNames = new List<string>()

{

"Name"

};

T entity = new T()

{

Name = "name"

};

Dictionary<string, string> requestData = (new RequestData<T> { Pager = pager, Entity = entity, Columns = columnNames }).Serialize();

FormUrlEncodedContent content = new FormUrlEncodedContent(requestData);

HttpResponseMessage response = myHttpClient.PostAsync("SelectPagerByColumns",content).Result;

* Get return - 获取调用结果：

RequestResult result = response.Content.ReadAsAsync<RequestResult>().Result;

TargetPager<T> targetPager = JsonConvert.DeserializeObject<TargetPager<T>>(entity.Data.ToString());

## SelectBySpecifyDate

* Call API in C# - 调用方式

HttpClient myHttpClient = new HttpClient();

string url = "http://ipAddress:port/ControllerName/";

myHttpClient.BaseAddress = new Uri(url);

HttpResponseMessage response = myHttpClient. GetAsync("SelectBySpecifyDate?id="+ id +"& timestamp="+ timeStamp).Result;

* Get return - 获取调用结果：

RequestResult result = response.Content.ReadAsAsync<RequestResult>().Result;

T entity = JsonConvert.DeserializeObject<T>(result.Data.ToString());

## SelectBySpecifyDate

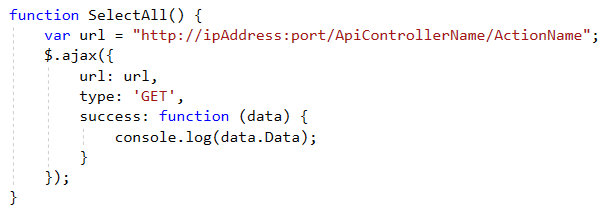
## SelectAllByDateTimeWithChild

# 前端调用Api

## SelectAll

Attention: ApiControllerName naming convention in following instructions : e.g. BlendChemicalApiController needs to be written as BlendChemicalApi in URL.

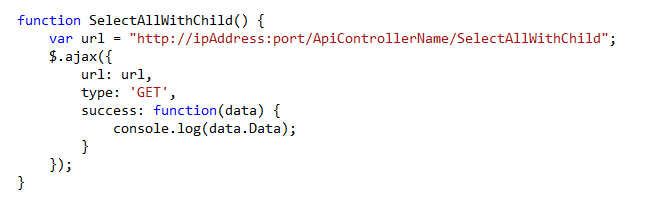
* Call API in Javascript - 调用方式



* Get return - 获取调用结果: data.Data;

## SelectAllWithChild

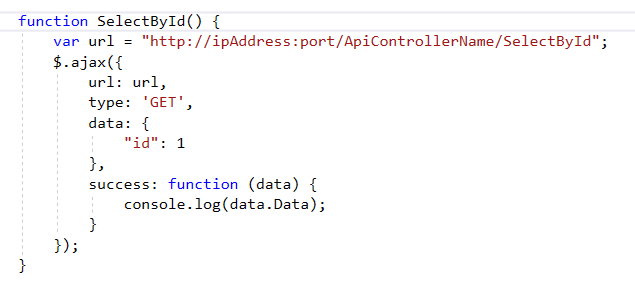
* Call API in Javascript - 调用方式



* Get return - 获取调用结果: data.Data;

## SelectById

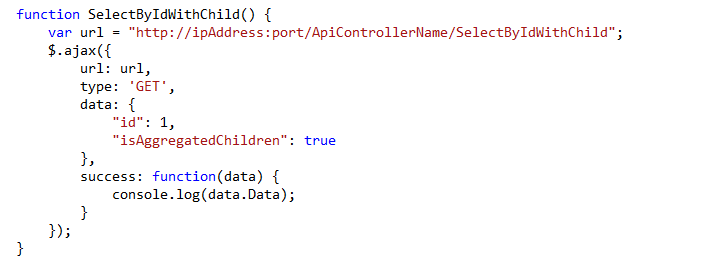
* Call API in Javascript - 调用方式



* 获取调用结果： data.Data;

## SelectByIdWithChild

* Call API in Javascript - 调用方式



* Get return - 获取调用结果: data.Data;

## SelectBy

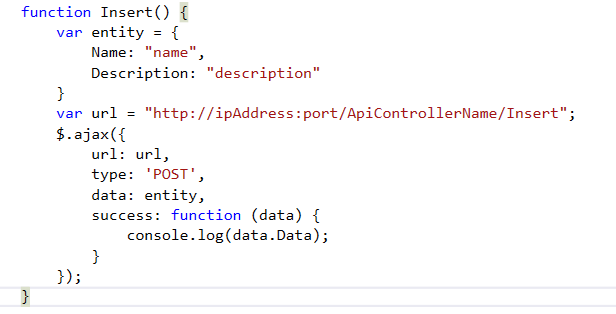
* Call API in Javascript - 调用方式



* Get return - 获取调用结果： data.Data;

## Insert

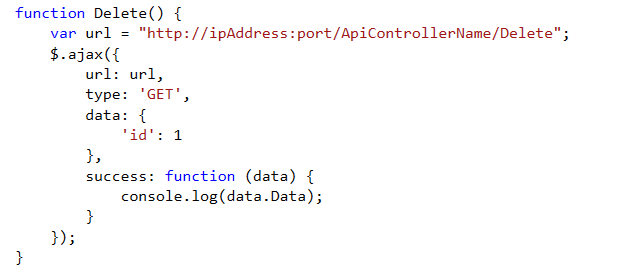
* Call API in Javascript - 调用方式



* Get return - 获取调用结果： data.Data;

## Delete

* Call API in Javascript - 调用方式



* Get return - 获取调用结果： data.Data;

## SelectPagerByColumns

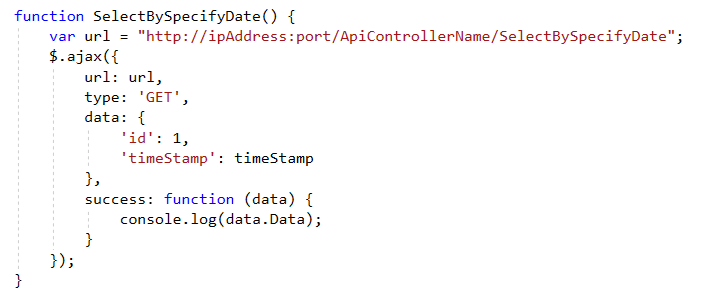
* Call API in Javascript - 调用方式

Same as 3.6. 采用同3.6 SelectBy同样的调用方式

* Get return - 获取调用结果： data.Data;

## SelectBySpecifyDate

* Call API in Javascript - 调用方式



* Get return - 获取调用结果： data.Data;

## SelectBySpecifyDate

## SelectAllByDateTimeWithChild