

Спецификация Proto (<https://protobuf.com/docs/introduction>) (json to proto: <https://json-to-proto.github.io/>)

```
syntax = "proto3";

package ecommerce;

// Сущности
message Product {
  string id = 1;
  string name = 2;
  string description = 3;
  double price = 4;
  int32 quantity = 5;
  repeated string category_ids = 6; // Бонус: в каких категориях присутствует продукт
}

message Category {
  string id = 1;
  string name = 2;
  repeated string product_ids = 3;
}

// Фильтры и пагинация
message ProductFilter {
  double max_price = 1;
}

message Pagination {
  int32 limit = 1;
  int32 offset = 2;
}

// Структуры запросов и ответов
message ProductRequest {
  Product product = 1;
```

```
ProductFilter filter = 2;
Pagination pagination = 3;
}

message ProductResponse {
  Product product = 1;
}

message CategoryRequest {
  Category category = 1;
}

message CategoryResponse {
  Category category = 1;
}

message AttachOrDetachRequest {
  string product_id = 1;
  string category_id = 2;
}

message ProductListResponse {
  repeated Product products = 1;
}

message CategoryListResponse {
  repeated Category categories = 1;
}

// Сервис
service Ecommerce {
  rpc AddProduct(ProductRequest) returns (ProductResponse);
  rpc EditProduct(ProductRequest) returns (ProductResponse);
  rpc DeleteProduct(ProductRequest) returns (ProductResponse);

  rpc AddCategory(CategoryRequest) returns (CategoryResponse);
  rpc EditCategory(CategoryRequest) returns (CategoryResponse);
}
```

```
rpc DeleteCategory(CategoryRequest) returns (CategoryResponse);

rpc AttachProductToCategory(AttachOrDetachRequest) returns (CategoryResponse);
rpc DetachProductFromCategory(AttachOrDetachRequest) returns (CategoryResponse);

rpc GetAllProducts(ProductRequest) returns (ProductListResponse);
rpc GetAllCategories(CategoryRequest) returns (CategoryListResponse);

rpc GetProductById(ProductRequest) returns (ProductResponse);
rpc GetCategoryById(CategoryRequest) returns (CategoryResponse);
}
```