RestSharp Tester

A simple CRUD example for a REST client communicating with a web API: <https://github.com/xnafan/RestSharpTester>

Here are the server- and client side codesamples which are used for communication.

|  |  |
| --- | --- |
| **SERVER** (ASP.NET Web API) *CustomerController.cs* Route: "api/v1/customers" | **CLIENT**  (REST client)  *CustomerRestClient.cs* |
| GET ALL | | |
| [HttpGet]  public ActionResult<IEnumerable<CustomerDto>?> Get()  {  return Ok(\_customerDao.GetCustomers()?.ToDtos());  } | public IEnumerable<CustomerDto>? GetCustomers()  {  //sends a GET request to "api/customers"  return \_client.Get<IEnumerable<CustomerDto>>(new RestRequest());  } |
| GET SINGLE | | |
| [HttpGet("{id}")]  public ActionResult<CustomerDto?> Get(int id)  {  var customer = \_customerDao.GetCustomerById(id);  if(customer == null) { return NotFound(); }  return Ok(customer.ToDto());  } | public CustomerDto? GetCustomerById(int id)  {  //sends a GET request to "api/customers/{id}"  var request = new RestRequest($"{id}");  return \_client.Get<CustomerDto>(request);  } |
| POST | | |
| [HttpPost]  public IActionResult Post([FromBody] CustomerDto customer)  {  \_customerDao.InsertCustomer(customer.FromDto());  return Ok();  } | public int InsertCustomer(CustomerDto customer)  {  //sends a POST request to "api/customers"  //with the Customer as a JSON object in body  //and returns the primary of the inserted customer  var request = new RestRequest("", Method.Post);  request.AddBody(customer);  return \_client.Post<int>(request);  } |

|  |  |  |
| --- | --- | --- |
| PUT | | |
| [HttpPut("{id}")]  public ActionResult Put(int id, [FromBody] CustomerDto customerToUpdate)  {  var customer = customerToUpdate.FromDto();  var customerFoundAndUpdated =\_customerDao.UpdateCustomer(customer);  if(!customerFoundAndUpdated) { return NotFound();}  return Ok();  } | public bool UpdateCustomer(CustomerDto customer)  {  //sends a PUT request to "api/customers"  //with the Customer as a JSON object in body  //and returns whether the customer was found on the server  var request = new RestRequest($"{customer.Id}");  request.AddBody(customer);  return \_client.Put<bool>(request);  } |
| DELETE | | |
| [HttpDelete("{id}")]  public ActionResult Delete(int id)  {  if(!\_customerDao.DeleteCustomer(id)) { return NotFound(); }  return Ok();  } | public bool DeleteCustomer(int id)  {  //sends a DELETE request to "api/customers/{id}"  //and returns the returnvalue  var request = new RestRequest($"{id}", Method.Delete);  return \_client.Delete<bool>(request);  } |
| SEARCH | | |
| //https://localhost:7288/api/v1/customers/search?partOfName=e&partOfEmail=h'  [HttpGet("search")]  public ActionResult<IEnumerable<CustomerDto>?> SearchCustomers(  [FromQuery] string partOfName,  [FromQuery] string? partOfEmail = null)  {  return Ok(\_customerDao.SearchCustomers(partOfName, partOfEmail)?.ToDtos());  } | public IEnumerable<CustomerDto>? SearchCustomers(  string partOfName, string? partOfEmail = null)  {  var request = new RestRequest("search")  .AddQueryParameter("partOfName", partOfName)  .AddQueryParameter("partOfEmail", partOfEmail ?? "");  return \_client.Get<IEnumerable<CustomerDto>>(request);  } |