

Chapter 1

The SERA side of the communication

PURPOSE

This manual describes the SERA-side of serving and processing data to and from a website. It has instructions how to set up the software and the directory structure involved and it gives samples of the data formats.

DIRECTORY STRUCTURE

There has to be a IN and OUT directory. The web-application has to put all the requests for information and processing in the IN directory. SERA places the answers in the OUT directory. It's highly recommended to keep a structure as follows:

<DRIVE>:\www\<>name>\api\sera\in and ..\out and ..\tmp

<name> stands for a company name, a database name or a purpose involved. It can be any name as long as there are no spaces and as long as it has a **maximum length of 8n characters!**

<DRIVE>: stands for any disk mapping at the server, where both SERA and the website have access to. Make sure that it's the **ONLY** part of the server or server farm that can be accessed by the website.

Sample: **E:\www\planning\api\sera\in** and ..\out and ..\tmp

This sample is continued in the other parts of this manual. Also is **disk M:** used as a sample disk mapping where SERA is running from.

FILES INVOLVED

Location	File name	Description
E:\www\planning\api\sera\	WEBSERV.EXE	The actual webserver that reads and writes from and to the SERA database. It is started by a request from the website.
E:\www\planning\api\sera\	SQLSERV.PAR	This file contains the information how to access the Sera database. See the standard Sera manual for more information. No adjustments required. Just a copy of M:\sera\paramet\sqlserv.par
E:\www\planning\api\sera	WEBREQ.BAT	This batch file calls a PHP script that forces the website to read ..\OUT\REQ*.JSON files. These files are containing automatically generated updates from customer, parts, bill of material data etc. from the Sera ERP database.
M:\sera\paramet	WEBLOCA.PAR	This file tells Sera ERP where to find the ..\in and ..\out directories. Note: use a structure like \\server\E\$\www\planning\api\sera or \\192.168.10.04\E\$\www\planning\api\sera

* Ask Sera for the latests updates.

JSON FILES

The communication goes by JSON files. To avoid errors with JSON files from other or not trusted resources SERA use NOT the .JSON extension but the **.JSN** extension.

Below are some samples. **Since each website represents the unique aspects of an organization, the fields mentioned can be different from your situation.**

WRITING TO THE IN DIRECTORY / READING FROM THE OUT DIRECTORY

The website has to write a *.JSN file into the E:\www\planning\api\sera\in directory. Please use unique file names like 10000002.JSN where a sequence number is increased. The length of the name should be max. 8 characters + the .JSN extension.

WEBSERV.EXE writes an answer with the same file name into E:\www\planning\api\sera\out.

After reading: **Delete the file** from E:\www\planning\api\sera\out

Note: use the ..\tmp directory to create files and only after writing move them to \in

SPECIAL JSON FILE TO STOP THE SERVER

If you place a file called STOP.JSN in the \IN directory, the webserver will stop immediately. The content of the file is irrelevant.

SAMPLE FILES

Note: these are the standards, but in most cases there are customer modifications!

GET STOCK LEVEL	
<pre>[{ "REQUEST": "STOCK", "Parts": [{ "PartID": "EW00410539", "CustomerID": "00016", "RequiredQuantity": "2000" }, { "PartID": "1000", "CustomerID": "00016", "RequiredQuantity": "1" }] }]</pre>	<pre>[{ "ANSWER": "STOCK", "Parts": [{ "PartID": "EW00410539", "ActualStock": "30.000", "AvailableToday": "604.000", "AvailableInOneDay": "604.000", "AvailableInTwoDays": "604.000", "RequiredQuantity": "2000.000", "RetailPrice": "25.000", "NetlPrice": "20.000", "DiscountPercentage": "20.00", "DeliveryDate": "2015-09-02" }, { "PartID": "1000", "ActualStock": "0.000", "AvailableToday": "0.000", "AvailableInOneDay": "0.000", "AvailableInTwoDays": "0.000", "RequiredQuantity": "1.000", "RetailPrice": "45.000", "NetlPrice": "27.500", "DiscountPercentage": "0.00", "DeliveryDate": "2199-12-31" }] }]</pre>

Note: the customer can be empty. Then the discount is zero and the net price is the same as the retail price.

GET SALES ORDERS PER CUSTOMER	
<pre>{ "REQUEST":"SALESORDERS", "CustomerID":"W0121" }</pre>	<pre>[{ "ANSWER": "SALESORDERS", "CustomerID": "W0121", "Headers": [{ "OrderID": "223456", "CustomerReference": "O5467", "OwnReference": "Website", "ContactPerson": "", "OrderDate": "2015-07-02", "CurrencyID": "KSH", "SalesArea": "N002", "PaytypeID": "CHE", "PayDays": "12", "DeliveryName": "Central warehouse", "DeliveryAddress": "Industrieweg 14", "DeliveryCity": "5145 JD Nuenen", "DeliveryCountry": "Nederland", "DeliveryTerms": "ENGEX", "Items": [{ "ItemNr": "0001000", "ItemType": "Part", "PartID": "EW00410539", "Description": "Seal hollow square, fill cap seal", "Quantity": "1.000", "DeliveryDate": "2015-07-31", "UnitPrice": "9.50", "RetailPrice": "0.00", "DiscountPercentage": "0.00" }, { "ItemNr": "0001010", "ItemType": "Part", "PartID": "SC008560", "Description": "Scully Groundbolt", "Quantity": "1.000", "DeliveryDate": "2015-07-31", "UnitPrice": "5.74", "RetailPrice": "0.00", "DiscountPercentage": "0.00" }] }, { "OrderID": "223458", "CustomerReference": "WEB 20150702", "OwnReference": "WEB 20150702", "ContactPerson": "", "OrderDate": "2015-07-02", "CurrencyID": "KSH", "SalesArea": "N002", "PaytypeID": "CHE", "PayDays": "12", "DeliveryName": "BMW Jansen", "DeliveryAddress": "Voortraat 4", "DeliveryCity": "2134 AD Katwijk Zh", "DeliveryCountry": "Nederland", "DeliveryTerms": "ENGEX",</pre>

	<pre>"Items": [{ "ItemNr": "0001000", "ItemType": "Part", "PartID": "EW00410539", "Description": "Seal hollow square, fill cap seal", "Quantity": "5.000", "DeliveryDate": "2015-08-14", "UnitPrice": "9.50", "RetailPrice": "0.00", "DiscountPercentage": "0.00" }] }</pre>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

POST SALES ORDERS INTO SERA	
<pre> { "REQUEST": "SALES", "Headers": [{ "SalesOrderHeaderID": "2276", "Name": "Sebastian Hogervorst", "Name2": "", "Address1": "Sterreschans 217", "City": "2411 BP LEIDEN", "CountryID": "NL", "WarehouseID": "", "Contact": "Mr.Jones", "CreateDate": "2016-05-23 11:24:57", "AreaID": "0000", "CustomerID": "10101", "CustomerReference": "My order 12345", "items": [{ "ItemNr": "0001", "Name": "RC6 12 V, chainwheel ", "PartID": "01.01.0001", "RetailPriceExclVAT": "44.5600", "RetailPriceInclVAT": "44.5600", "SalesCurrencyPriceExclVAT": "44.5600", "SalesCurrencyPriceInclVAT": "44.5600", "DiscountPercentage": "0.00", "SalesCurrencyAmountExclVAT": "89.12", "SalesCurrencyAmountInclVAT": "89.12", "Quantity": "2.000", "DeliveryDate": "2016-06-14" }, { "ItemNr": "0002", "PartID": "13.01.0005", "SalesCurrencyPriceExclVAT": "50.0000", "Quantity": "2.000", "DeliveryDate": "2016-06-04" }] }] } </pre>	<pre> [{ "ANSWER": "SALES", "Headers": [{ "SalesOrderHeaderID": "223456", "Message": "Posting done", "Items": [{ "ItemNR": "1000", "Message": "Posting done" }, { "ItemNR": "1010", "Message": "Posting done" }] }, { "OrderID": "223458", "Message": "Posting done", "Items": [{ "ItemNR": "1000", "Message": "Posting done" }] }] }] </pre>

POST QUOTATIONS INTO SERA	
<pre>[{ "REQUEST": "QUOTATION", "Headers": [{ "QuotationHeaderID": "2276", "Name": "Sebastiaan Hogervorst", "Name2": "", "Address1": "Sterreschans 217", "City": "2411 BP LEIDEN", "CountryID": "NL", "Phone": "065470199", "Email": "sales@jansen.com", "Contact": "Mr.Jones", "CreateDate": "2016-05-23 11:24:57", "AreaID": "0000", "CustomerID": "10101", "CustomerReference": "My quotation 12345", "items": [{ "ItemNr": "0001", "Name": "RC6 12 V, chainwheel ", "PartID": "01.01.0001", "RetailPriceExclVAT": "44.5600", "RetailPriceInclVAT": "44.5600", "SalesCurrencyPriceExclVAT": "44.5600", "SalesCurrencyPriceInclVAT": "44.5600", "DiscountPercentage": "0.00", "SalesCurrencyAmountExclVAT": "44.56", "SalesCurrencyAmountInclVAT": "44.56", "Quantity": "1.000", "DeliveryDate": "2016-06-14" }, { "ItemNr": "0002", "PartID": "13.01.0005", "SalesCurrencyPriceExclVAT": "50.0000", "Quantity": "2.000", "DeliveryDate": "2016-06-04" }] }] }]</pre>	<pre>[{ "ANSWER": "QUOTATION", "Headers": [{ "QuotationHeaderID": "223456", "Message": "Posting done", "Items": [{ "ItemNR": "1000", "Message": "Posting done" }, { "ItemNR": "1010", "Message": "Posting done" }] }, { "QuotationID": "223458", "Message": "Posting done", "Items": [{ "ItemNR": "1000", "Message": "Posting done" }] }] }]</pre>

GET CUSTOMER DATA	
<pre>{ "REQUEST": "CUSTOMERDATA", "CustomerID": "W0121" }</pre>	<pre>{ "ANSWER": "CUSTOMERDATA", "CustomerID": "W0121", "HeadOfficeCustomerID": "W0121", "Name": "ABDIFATAH KAPUA EBEI", "Address1": "P.O. BOX", "City": "LOKITAUNG", "CountryID": "Kenya", "Phone": "0735-236959", "ArealID": "N002", "Email": "JENSEN@TEST.COM", "Fax": "", "Contact": "", "PaymentDays": "12", "DeliveryTermID": "ENGEX", "Accounting": "", "CurrencyID": "KSH", "CreditLimitAmount": "1000000.00", "VatTypeID": "EX", "VatNumber": "", "BankAccount": "NL59INGB0673015610", "BankName": "INGNL2A" }</pre>

GET A NEW CUSTOMER NUMBER	
<pre>{ "REQUEST": "NEWCUSTOMER" }</pre>	<pre>{ "ANSWER": "NEWCUSTOMER", "CustomerID": "10121" }</pre>

GET OPEN AMOUNTS PER CUSTOMER	
<pre>{ "REQUEST": "OUTSTANDINGAMOUNTS", "CustomerID": "W0121" }</pre>	<pre>[{ "ANSWER": "OUTSTANDING", "CustomerID": "W0121", "CurrencyID": "KSH", "Invoices": [{ "InvoiceID": "000152", "Date": "07-04-2015", "Description": "FAKTUUR", "OutstandingAmount": "500.00", "OriginalAmount": "500.00", "PaymentDate": "2015-04-19" }, { "InvoiceID": "000153", "Date": "07-04-2015", "Description": "FAKTUUR", "OutstandingAmount": "600.00", "OriginalAmount": "600.00", "PaymentDate": "2015-04-19" }, { "InvoiceID": "000158", "Date": "29-07-2015", "Description": "FAKTUUR", "OutstandingAmount": "4131.00", "OriginalAmount": "4131.00", "PaymentDate": "2015-08-10" }, { "InvoiceID": "000160", "Date": "29-07-2015", "Description": "FAKTUUR", "OutstandingAmount": "1377.00", "OriginalAmount": "1377.00", "PaymentDate": "2015-08-10" }] }]</pre>

POST PURCHASE ORDERS INTO SERA	
<pre>[{ "REQUEST": "PURCHASE", "Headers": [{ "OrderID": "323456", "SupplierID": "2010", "Reference": "Website", "DeliveryName": "Central warehouse", "DeliveryAddress": "Industrieweg 14", "DeliveryCity": "5145 JD Nuenen", "DeliveryCountry": "Nederland", "WarehouseID": "BU", "Added": "2015-07-02 11:27:45", "Items": [{ "ItemNr": "1000", "PartID": "EW00410539", "UnitPrice": "9.50", "DeliveryDate": "2015-07-31", "Quantity": "1" }, { "ItemNr": "1010", "PartID": "SC008560", "UnitPrice": "5.74", "DeliveryDate": "2015-07-31", "Quantity": "1" }] }], { "OrderID": "323457", "SupplierID": "2020", "Reference": "Website", "SupplierName": "Chaffic Oudja", "DeliveryName": "Volvo Oostende", "DeliveryAddress": "Hofstedestraat 26", "DeliveryCity": "Oostende", "DeliveryCountry": "Belgie", "WarehouseID": "DS", "Added": "2015-07-02 11:27:45", "Items": [{ "ItemNr": "1000", "PartID": "EW00410539", "UnitPrice": "9.50", "DeliveryDate": "2015-07-31", "Quantity": "1" }, { "ItemNr": "1010", "PartID": "SC008560", "UnitPrice": "5.74", "DeliveryDate": "2015-07-31", "Quantity": "2" }] }] }]</pre>	<pre>[{ "ANSWER": "PURCHASE", "Headers": [{ "OrderID": "323456", "Message": "Posting done", "Items": [{ "ItemNR": "1000", "Message": "Posting done" }, { "ItemNR": "1010", "Message": "Posting done" }] }, { "OrderID": "323457", "Message": "Posting done", "Items": [{ "ItemNR": "1000", "Message": "Posting done" }, { "ItemNR": "1010", "Message": "Posting done" }] }] }]</pre>

POST HOURS INTO SERA	
<pre>[{ "REQUEST": "HOURS", "Items": [{ "ScheduleDetailID": "1", "EmployeeID": "SES", "Date": "2015-07-24", "Start": "2015-07-24 08:45:00", "End": "2015-07-24 09:45:00", "WorkTypeID": "MAP", "Description": "Welding", "InvoiceHours": "01:00:00", "WorkedHours": "01:00:00", "Percentage": "100", "OrderType": "1", "OrderID": "100198" }, { "ScheduleDetailID": "8", "EmployeeID": "SES", "Date": "2015-07-23", "Start": "2015-07-23 09:00:00", "End": "2015-07-23 11:00:00", "WorkTypeID": "PRG", "Description": "Programming", "InvoiceHours": "02:00:00", "WorkedHours": "02:00:00", "Percentage": "100", "OrderType": "1", "OrderID": "100104" }] }]</pre>	<pre>[{ "ANSWER": "HOURS", "Items": [{ "ScheduleDetailID": "1", "Message": "Posted" }, { "ScheduleDetailID": "8", "Message": "WorkTypeID unknown" }] }]</pre>

CREATE OR UPDATE CUSTOMERS IN SERA	
<pre>[{ "REQUEST": "CUSTOMERUPDATE", "Customers": [{ "CustomerID": "G0001", "Name": "Eerste klant", "DeliveryAddress": "Industrieweg 14", "DeliveryCity": "5145 JD Nuenen", "DeliveryCountry": "Nederland", "CurrencyID": "EUR", "PayTerms": "60" }, { "CustomerID": "G0002", "Name": "Tweede klant", "ABC": "22222", "LineID": "200", "DeliveryAddress": "Industrieweg 14", "DeliveryCity": "5145 JD Nuenen", "DeliveryCountry": "Nederland", "CurrencyID": "EUR", "PayTerms": "60" }, { "CustomerID": "G0003", "Name": "Derde klant", "ArealD": "WEB", "DeliveryAddress": "Industrieweg 14", "DeliveryCity": "5145 JD Nuenen", "DeliveryCountry": "Nederland", "CurrencyID": "USD", "PayTerms": "90" }] }]</pre>	<pre>[{ "ANSWER": "CUSTOMERUPDATE", "Customers": [{ "CustomerID": "G0001", "Message": "Updated" }, { "CustomerID": "G0002", "Message": "Added" }, { "CustomerID": "G0003", "Message": "Updated ArealD unknown. Changed to 0000" }] }]</pre>

Available fields (all optional except the CustomerID field):

HeadCustomerID	Country
SearchCode	Name (60 chars long split over cuname and cuname2)
ArealD	Address
City	DeliveryAddress
DeliveryCity	LanguageID
ContactPerson	Phone
WebPassword	Email
Website	Fax
Administration	MailCode
SelectionCode	LineID
CurrencyID	DeliveryTermID
PaymentDays	CreditLimit
PayTypeID	VatTypeID
VatNumber	BankAccount
BankName	

CREATE OR UPDATE PARTS IN SERA	
<pre>[{ "REQUEST": "PARTUPDATE", "Parts": [{ "PartID": "G0001", "Name": "Eerste artikel", "SupplierID": "1000", "CurrencyID": "EUR", "PayTerms": "60" }, { "PartID": "G0002", "Name": "Tweede artikel", "SupplierID": "1000", "CurrencyID": "EUR", "PayTerms": "60" }, { "PartID": "G0003", "Name": "Derde artikel", "SupplierID": "2000", "PayTerms": "90" }] }]</pre>	<pre>[{ "ANSWER": "PARTUPDATE", "Parts": [{ "PartID": "G0001 ", "Message": "Updated" }, { "PartID": "G0002 ", "Message": "Updated" }, { "PartID": "G0003 ", "Message": "Updated" }] }]</pre>

Available fields (all optional except the PartID field):

- | | |
|------------------------|------------------------------------------|
| PartGroupID | Name |
| Name2 | Barcode |
| Unit | ObsoletePartYN |
| SupplierID | SupplierPartNumber |
| DeliveryDays | CurrencyID |
| PurchasePrice | RetailPrice |
| RetailPrice2 | ExtendedText (line split is \r\n) |
| Weight | IntegerQuantitiesYN |
| PackingQuantity | |

GET STOCK MUTATIONS	
<pre>[{ "REQUEST": " STOCKMUTATIONS ", "FromDate": " 2016-02-02 " }]</pre>	<pre>[{ "ANSWER": "STOCKMUTATIONS", "Details": [{ "PartID": "0001097", "Date": "2016-03-20", "Time": "12:27:29", "Reference": "100012-100019", "Type": "Invoiced", "Quantity": "-1" }, { "PartID": "0000219", "Date": "2016-02-02", "Time": "02:12:03", "Reference": "100001", "Type": "Received", "Quantity": "2" }] }]</pre>

UPDATE WEB FROM SERA-ERP

CUSTOMER UPDATE

```
{
  "UPDATEREQUEST": "Customer",
  "CustomerID": "W0121",
  "HeadOfficeCustomerID": "W0121",
  "Name": "ABDIFATAH KAPUA EBEI",
  "Address1": "P.O. BOX",
  "City": "LOKITAUNG",
  "CountryID": "Kenya",
  "Phone": "0735-236959",
  "ArealID": "N002",
  "Email": "JENSEN@TEST.COM",
  "Fax": "",
  "Contact": "",
  "PaymentDays": "12",
  "DeliveryTermID": "ENGEX",
  "Accounting": "",
  "CurrencyID": "KSH",
  "CreditLimitAmount": "1000000.00",
  "VatTypeID": "EX",
  "VatNumber": "",
  "BankAccount": "NL59INGB0673015610",
  "BankName": "INGNL2A",
  "ActiveYN", "1"
}
```

PART UPDATE

```
{
  "UPDATEREQUEST": "Part",
  "PartID": "01.01.011",
  "Name": "Hydraulic hose ¾ inch 3 meter",
  "Description": "<long text>",
  "Barcode": "1234567890123",
  "Unit": "Piece",
  "MaximumFromStock": "99999",
  "ObsoletePartYN": "0",
  "LeadTimeDays": "14",
  "SupplierID": ""
}
```

PARTCATEGORY UPDATE

```
{
  "UPDATEREQUEST": "PartCategory",
  "PartCategoryID": "010302",
  "Name": "Hoses",
  "ActiveYN", "1"
}
```

PART_PARTCATEGORY UPDATE

```
{
  "UPDATEREQUEST": "Part_PartCategory",
  "PartCategoryID": "010302",
  "PartID": "01.01.0011",
  "ActiveYN", "1"
}
```


BILL OF MATERIAL UPDATE

```
{  
  "UPDATEREQUEST": "BillOfMaterial",  
  "ParentPartID": "01.01.0011",  
  "PartID": "04.05.0052",  
  "PositionNumber": "14",  
  "Route": "0",  
  "Quantity": "1",  
  "ActiveYN": "1"  
}
```

Chapter 2

The web side of the communication

Introduction

Purpose

This manual describes the WEB-side of serving and processing data to and from your website. It had instructions how to set up the software and the directory structure involved and it gives samples of the data formats. Also there is some PHP example code to get you started.

REST API

The SERA API is based on a REST API library. If you have any experience with REST, this should be really easy for you to implement. If not, don't worry, the example codes we provide are functional. You can copy them and start working immediately.

Installation

Webserver

You have to install a webservice on a server in your network that can communicate with the server where SERA is running. This webservice will be handling all the request that you do from your website.

We advise to use XAMPP. This is a free webservice running Apache with PHP included. You can run any kind of webservice you want, as long as it is running PHP and has .htaccess support.

Website

Your website will communicate with the server running the new webservice. You have to create a VHOST and update your DNS settings so your API is accessible over the internet.

For example; we see a directory structure example like the one below:



Your VHOST should have the API folder as the document root. This means that if you add a VHOST like api.serasoft.com, it will run the scripts in that folder.

The API structure

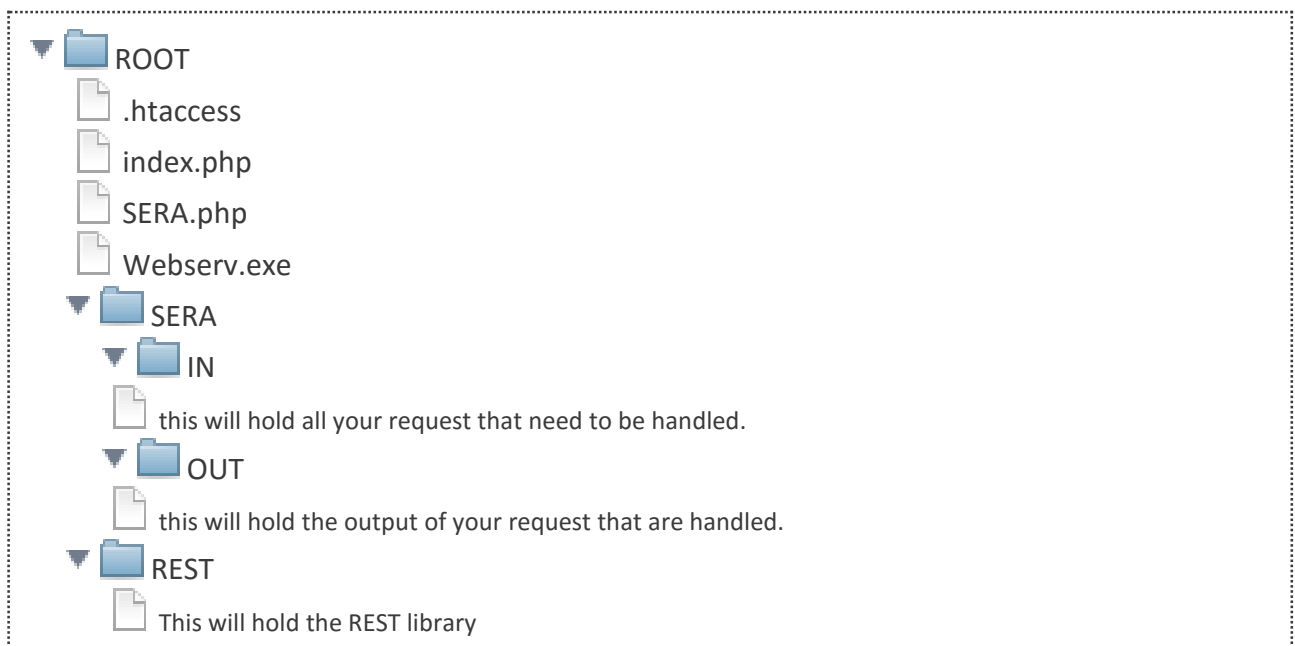
Once you've created all the directories and installed the webserver as explained, you can start working on the API. We need a couple of folders and files to make it work.

We will be using the document structure below and we will call the api folder "ROOT".

ROOT =  E  www  planning  api

Create the following folder structure:

* Please note that there will be more files in this folder as explained in the "sera to web manual". We don't explain those files here.



The API files

Introduction

In this chapter we will show the code of the files in the ROOT directory. All the files in the REST folder are not explained. You don't have to edit them or do anything to them to make the API work. We advise you to leave them as they are.

.htaccess

* Alterations to this file should not be necessary.

```
DirectoryIndex index.php
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteRule ^$ index.php [QSA,L]
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteRule ^(.*)$ index.php [QSA,L]
</IfModule>
<IfModule mod_php5.c>
    php_flag display_errors Off
</IfModule>
```

.htaccess

index.php

* Only change the ROOT variable in this document.

```
<?php
/* Change this to your own structure */
$ROOT = 'E:/www/planning/api/';

require $ROOT . 'rest/server/vendor/restler.php';
use Luracast\Restler\Restler;

$r = new Restler ();

/* This will add the SERA.php file to the API */
$r->addAPIClass('SERA');
$r->handle();

?>
```

index.php

SERA.php

This file will create the .JSON file. No alterations are required, however, if you want to do some custom handling of information in the API you can do that in this file. By default, this file will return the response of SERA or it will let you know that the SERA is not running correctly.

This file will also try to start up STARTWEB.EXE if it detects that its not running. If this fails, it will let you know.

Using the API

Introduction

In this chapter we will show you PHP example code to do a request to your new SERA API you've just setup. This is really straight forward and is done with very little code.

In this example we assume that the API is running on `api.serasoft.com`

```
<?php
```

```
example.php
```

```
$url = 'http://api.serasoft.com/SERA/handle';

$fields = array (
    'Request' => 'CUSTOMERDATA',
    'CustomerID' => 'W0121'
);

$fields_string = "";

foreach($fields as $key => $value) {
    $fields_string .= $key . '=' . urlencode($value) . '&';
}

rtrim($fields_string, '&');

$ch = curl_init();
curl_setopt($ch,CURLOPT_URL, $url);
curl_setopt($ch,CURLOPT_POST, count($fields));
curl_setopt($ch,CURLOPT_POSTFIELDS, $fields_string);
curl_setopt($ch,CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch,CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch,CURLOPT_TIMEOUT_MS, 3000);

$result = json_decode(curl_exec($ch));
curl_close($ch);
```