

The power of API: Giving your customers that little extra

JUNE 19, 2020 | 4 MIN – READ

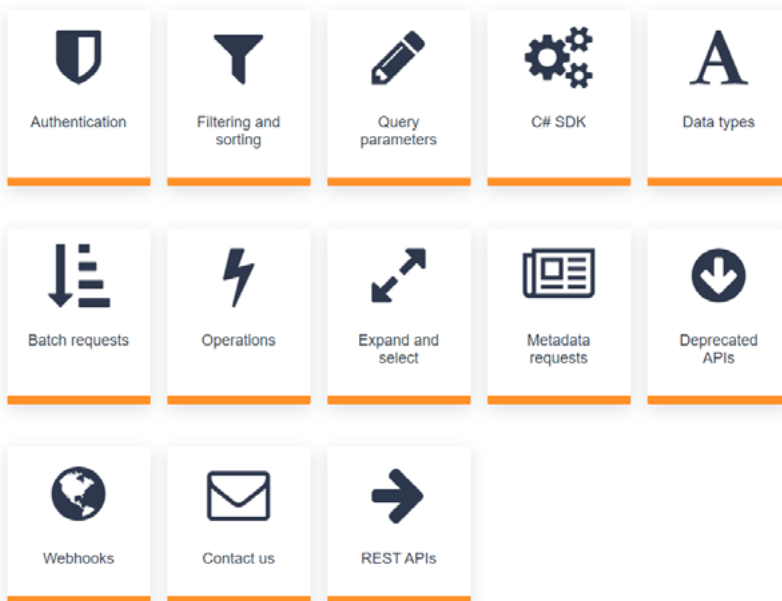
No ERP system will fit perfectly with all business types. A modern ERP system needs to be robust, flexible and configurable. RamBase is not only an ERP system, but an ERP platform. With platform technology you will have an unlimited world of possibilities with your system, since you can integrate information from other software systems, BI, external websites, customer portals, your machine park etc. The integration job is all made possible with API technology. This article will show you how RamBase, together with competent partners, can solve process needs by delivering a standard ERP system with a few key customized applications.

The possibilities with API in RamBase

An API, short for Application Programming Interface, is a set of routines, protocols, and tools for building software applications. RamBase offers partners, and IT-interested customers, a platform for developing RamBase applications based on RESTful API. A RESTful API uses HTTP requests to GET, PUT, POST and DELETE data and provides a great deal of flexibility for creating web services.

On RamBase.com the API resources are available for everybody to examine via the open link: api.ramse.net. This site offers a catalog with all RamBase API's for developers to build apps, extensions, and open integrations with RamBase.their products and meeting customer's real needs. Nobody wins by offering a system that do not fit and no one uses.

Also, the solution you may think is the best based on a few internet searches, may not be the perfect fit when you look under the hood. To make sense of it all, here are some main ways to group the different system options.



THE FRONTPAGE FOR RAMBASE API RESOURCE CATALOG

To help developers design applications with the look and feel consistent with the RamBase brand, the [RamBase Guidelines](#) site is a useful tool. Besides offering design guidelines, this site presents information about setup environment, creating applications, building components and list of functional User Interface (UI) elements. RamBase recommend using Visual Studio Code editor when coding, but any text editor can be used. With VS Code you will get access to RamBase code snippets and typescript intelligence.



THIS SITE OFFERS USEFUL DESIGN GUIDELINES FOR RAMBASE APPLICATIONS

Developing an application for LogiTrans

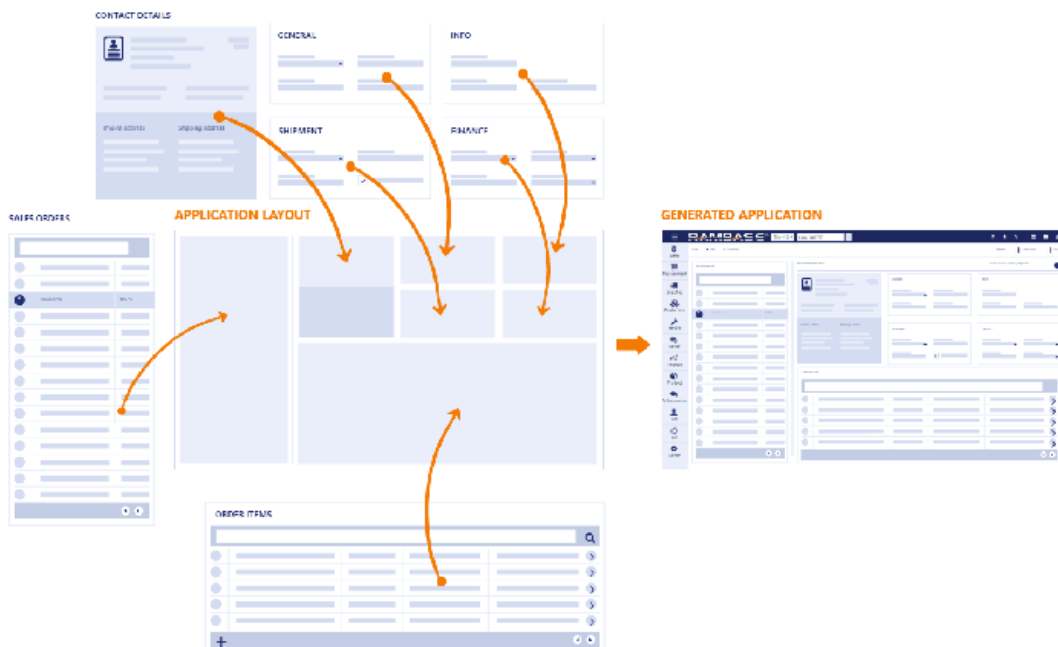
In the spring of 2020 LogiTrans became a new customer to RamBase. They wanted a system that delivered standard functionality, since their previous experience was with highly customized systems. LogiTrans teamed up with RamBase partner BDO to look into the potential with the RamBase ERP platform. Together they examined LogiTrans's key processes and how they would fit with RamBase. The standard functionality in RamBase was a good match for LogiTrans to manage their logistic processes.

However, no system will fit 100%. This was also the case for LogiTrans and RamBase. The main business process for LogiTrans is not storage, but they often have customers with need for remote storage. To avoid potential legal disputes about the conditions of the delivered items LogiTrans needed to document the condition of the merchandise with photos. The solution was to let their employees take pictures with iPad or smartphones of the delivered items and quickly add the picture in RamBase together with the inventory item. The standard functionality in RamBase for inventory management did not originally allow for simply documenting inventory flow, so this had to be developed for the customer.

IMAGE UPLOADER VIEWER

Using the well-established RamBase guidelines and API catalog, BDO engaged Solv to assist with their technical expertise for developing the code for the new application Image uploader viewer for LogiTrans.

The Image uploader viewer is a new component that is part of a larger application window. A component can be reused in several different applications.



THE ILLUSTRATION SHOWS 10 COMPONENTS ADDED TO AN APPLICATION

To create the image uploader viewer the developers utilized the functionality within the RamBase REST API framework. All the different modules in RamBase have associated API's down to application- and component level. For this new component, the developers used relevant information under Logistics – Goods reception – Item.

The screenshot shows the RAMBASE API Resources interface. The top navigation bar includes the RAMBASE logo, 'API Resources', and links for 'APIs', 'Getting started', and 'My client'. A search bar is present. On the left, a sidebar lists various modules: Auction, Base, Collaboration, CRM, Finance, Human resource, Logistics (highlighted), Procurement, Product, Production, Quality assurance, Rental, Sales, Service, and System. The main content area is titled 'Goods reception' and shows a breadcrumb trail 'Logistics > Goods reception'. Below this, there's a 'Tags and resources' section with a 'All tabs' dropdown. It lists three resources: 'Assignment', 'Item', and 'Notification'. Each resource has a corresponding API endpoint and method: 'logistics/goods-receptions' (GET), 'logistics/goods-receptions' (POST), and 'logistics/goods-receptions/{goodsReceptionId}' (GET).

THE API RESOURCE CATALOG IS SORTED BY MODULES AND THEN FEATURES IN RAMBASE

The data for coding the component is presented in either Json- or XML format and can be found under the output-tab for the relevant API. The developers edited and added to the code in VS-editor to deliver the new functionality that LogiTrans needed.

This screenshot shows the 'GET logistics/goods-reception-items' endpoint page. The breadcrumb trail is 'Logistics > Goods reception > Item > GET logistics/goods-reception-items'. The 'Output' tab is selected, showing the response format. The response is in JSON format, as indicated by the 'Format' dropdown. The JSON structure is as follows:

```

1 {
2   "goodsReceptionItems": [
3     {
4       "goodsReceptionItemId": "(Integer)",
5       "status": "(Integer)",
6       "createdAt": "(DateTime)",
7       "lineNumber": "(Integer)",
8       "suppliersReferenceNumber": "(String)",
9       "suppliersProductName": "(String)",
10      "hasCustomizedStructure": "(Boolean)",
11      "quantity": "(Decimal)",
12      "heatNumber": "(String)",
13      "availableQuantity": "(Decimal)",
14      "stockQuantity": "(Decimal)",
15      "remainingQuantity": "(Decimal)",
16      "productionLot": "(String)",
17      "bestBeforeDate": "(Date)",
18      "price": {
19        "currency": "(String)",
20        "grossPrice": "(Decimal)",
21        "netPrice": "(Decimal)"

```

THE CODE FOR "GOODS RECEPTION ITEM" IN JSON FORMAT.

The finished component is uploaded to the app catalog in RamBase and implemented into the customers goods reception application.

The result: a unique feature for LogiTrans to manage their logistic process, while still using all the standard functionality in RamBase.

THE STANDARD GOODS RECEPTION VIEW FOR ALL CUSTOMERS

THE GOODS RECEPTION VIEW FOR LOGITRANS WITH ADDED COMPONENT "IMAGE UPLOADER VIEWER"

THE FINISHED FEATURE IN USE

The collaboration between LogiTrans, BDO and Solv, using the available resource in the RamBase API platform, made it possible for LogiTrans to have a key feature in their system that has a great impact on their day to day operations.