Visit's API

The main goal of the API is to facilitate automated communication - initiated from 3rd party systems / software packages - with the Visit Database, mainly focused on creating, reading, updating and deleting visitor and partner information.

The API can be used for many reasons such as:

- Extracting the registration data for use in 3rd party applications, such as data warehouses, housing/booking systems, dynamic websites, accounting systems;
- Automated management of exhibitor details/accounts in the Visit system;
- Automated management of visitor's personal records in order to pre-populate the registration system / forms or to create registrations from a 3rd party system.

How does it work?

- The API is a link between the Visit system and third party applications. You decide what you want to know or update. You give the commands to the API and Visit sends an answer back to your application with the most up to date information.
- Visit empowers you to take full control over the integration. Once your account is created, you can set up you API keys and decide what to read, create, update and delete.
- There are 2 types of API keys available: at organisation level or event level.
- All communication is processed via **http get, post, put** and **delete** messages.

RESTful API

The API is a so-called RESTful API (Representational State Transfer). REST-style architectures conventionally consist of clients (you) and servers (Visit). Clients initiate requests to servers; servers process requests and return appropriate responses. Requests and responses are built around the transfer of representations of resources. A resource can essentially be any coherent and meaningful concept that may be addressed. A representation of a resource is typically a document that captures the current or intended state of a resource. The client begins sending requests when it is ready to make the transition to a new state. While one or more requests are outstanding, the client is considered to be in transition. The representation of each application state contains links that may be used the next time the client chooses to initiate a new state-transition.

What sort of information can I send or receive?

Visitor & Partner Information

- Contacts
- Visitors/Visitor profiles
- Partners
- Partner profiles
- Orders

Setup Information

- Events
- Questions
- Registration Types
- Registration Forms

Two Ways to Implement Data Retrieval

Using
change notifications

Using revision

For more information check out this page: **https://api.n200.com/console/**