



# Open API recommendations for cities

By the six largest cities in Finland

**6Aika**

Leverage from  
the EU  
2014–2020



European Union  
European Regional  
Development Fund  
Euroopan Sosiaalinen  
Fundi

**Publisher**

The 6Aika Open Data and Interfaces Spearhead Project  
Helsinki  
Espoo  
Vantaa  
Tampere  
Turku  
Oulu

**Editor-in-chief**

Annukka Varteva

**Text**

© Forum Virium Helsinki, 2016  
User licence: CC BY-SA 4.0

**Illustrations and layout**

© Paper Planes Oy, 2016  
User licence: CC BY-ND 4.0

These materials can be used as long as the publisher is credited. The attribution must be removed if so requested by the publisher of the material.

The aim of the 6Aika Open Data and Interfaces Spearhead Project is to bring the opening of data as part of the cities' normal operations and to help the cities facilitate the creation of data-based services and business.

**Coordinator**

Matti Saastamoinen

# Content

<b>Purpose of these guidelines</b>	<b>4</b>
<b>Vision: Towards API-driven city services</b>	<b>7</b>
Benefits for cities	8
Harmonised APIs	8
<b>Shared principles</b>	<b>10</b>
Working for openness in API development and use	10
Facilitating discovery and use	11
Building on API standards and technical guidelines	11
Ensuring API-related competence	11
Making open APIs a part of cities' strategic policies	12
Developing management models for open API standards and products	12
Strengthening the collaboration between local, national and international actors	13
<b>City-specific measures</b>	<b>14</b>
Support the definition of an open API	14
Create a management model for the APIs	18
Use open licences wherever possible	19
Harmonise the data format, data model or API	19
Aim for a sufficient SLA	21
Favour open tools	21
Create instructions and distribute them openly	21
Arrange training related to open APIs	22
<b>Checklist for opening an open API</b>	<b>23</b>

# Purpose of these guidelines



4

Mostly due to accelerating digitalisation, APIs are rapidly becoming increasingly important from a technological and service-oriented standpoint. Indeed, a need has emerged for a document that describes the importance of APIs and the goals related to them from the cities' perspective. These recommendations are intended as a general policy to be applied alongside the strategic policies, principles and guidelines that often guide ICT and service development in cities.

This document also presents the consolidated view of the six largest cities in Finland – Helsinki, Espoo, Vantaa, Tampere, Turku and Oulu – on how to develop open APIs through inter-city cooperation.

The document is divided into three chapters. The chapter *Vision* describes the joint vision of the Six Cities on how to develop API-driven city services. The chapter *Shared principles* describes the objectives of the cities on a general level. The chapter *City-specific measures* describes the measures aimed at reaching the goals in a more concrete fashion.

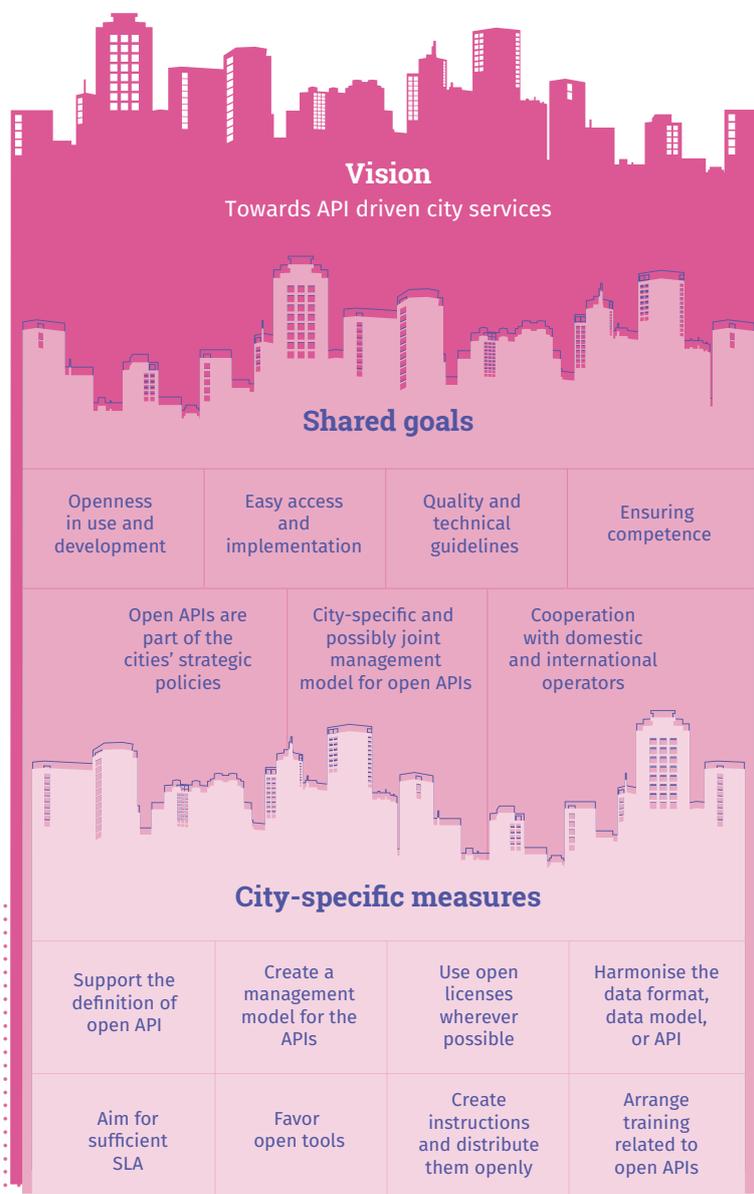
In order to facilitate practical work and support recommendations, more specific practical instructions and guidelines as well as examples of the measures that have been found effective in API harmonisation between the cities will also be prepared. The upper-level recommendations in this document are intended to serve as more permanent operating models.

An open API is an application programming interface, the properties of which are all public, that can be used without restricting terms and conditions. Open APIs are free to use and the users are not required to ask the permission of the API provider or indicate the intended purpose of using the API in advance.

The API recommendations can be used in the cities' strategic policies.



The vision can be achieved with the help of shared goals and city-specific measures.



# Vision: Towards API-driven city services

Cities constantly produce massive quantities of information that is eligible for public access. In order to maximise the utilisation of this great public resource, it is essential to open up public data so that it is machine-readable and permits reuse – in other words, as open data.

It is advisable to publish data through open APIs especially when similar data is being published by multiple cities. Harmonisation of open APIs enables the utilisation of similar data collected from various sources and the creation of scalable applications on top of the APIs.

The aim of the Six Cities is to operate openly and publish public information produced in their systems as open data, unless there is reason to do otherwise. In the development of city services, the utilisation of open APIs is incorporated into the plans at an early stage whenever possible.

Open data is machine-readable and can be utilised by anyone free of charge.

Open APIs facilitate the joint use of data.

## Benefits for cities

Cities can greatly benefit from using harmonised open APIs. For example, they can be used to develop cities' own operational processes and to

The public information produced by cities is published as open data, unless something prevents it.

achieve cost savings. When the data produced by a city is made easily available, it can also be used by departments other than the organisation that produced the data. In addition to this, open APIs prevent situations where data cannot be easily retrieved from a system and must, instead, be ordered separately from the system vendor – often for an additional charge. An open API can also be implemented in such a way that it compiles data automatically from a variety of sources and delivers them to the user in a unified format.

Open APIs can be used to develop operational processes and to achieve cost savings.

When a city organisation is using an open API in its own operations, any possible shortcomings and development suggestions are brought up faster and the maintenance operations are more reliable. Similarly, when actors inside and outside the city organisation are using the same open API, the city avoids having to maintain two overlapping systems.

## Harmonised APIs

In many cases, it is easier for users of open data if data gained from a variety of sources is published through open APIs. For example, open, standardised and harmonised APIs enable developers to gather similar data from multiple sources and cities irrespective of their background systems.

When data is published through an open API, the developer does not necessarily need to be provided access to the background system itself.

The API is also used to ensure that the background system is not stressed excessively.

By creating applications based on open data developers can help cities offer better services for citizens.

However, it may sometimes be difficult to unify the data content. In such cases, it is still beneficial to provide data from a variety of sources through harmonised APIs in a consistent data format and within a shared data model.

Open APIs are used to provide data users with opportunities for the development of new business, services and applications. As a result, the primary external user of an open API is usually a developer or developer company that provides services to its own customer base.

The need for an API is resolved on a case-by-case basis.

In order to ensure that data of the same type is as easy as possible to utilise, standards are required to define the data format, data structure and APIs. Globally established standards should be used, since data and open APIs are also developed on an international level.

It should be noted that the choice of an API type always affects the ways in which the data can be used. Among other things, the interface determines what types of searches can be performed on the materials. Although an open API usually facilitates data usage, the data should also always be provided as a static file, exported from the database, where possible.

It is not always best practice to publish all data through an API. Instead, the type of and need for an API depend on the nature and scope of the material that is being opened up. The need for an API is always resolved on a case-by-case basis. For example, it may not be necessary to use an API to publish materials that are limited in scope or updated only once a year, as in these cases a structurally formatted file (such as a CSV table, XML file or JSON file) may suffice.

# Shared principles

10

## Working for openness in API development and use

Cities should operate openly and make the public information produced within their systems available as open data, unless there is specific reason to do otherwise. City APIs should be designed to be open wherever possible. Cities can benefit from the insights companies, developer communities, and individual developers can offer, so, if possible, the targeted users of an open API should be involved early on in the design and development stages. The wider the variety of views that are taken into account, the better the open API will meet the corresponding needs. If city employees will

be utilising the API, it is useful to get them on board too.

To make sure the information or service has diverse use applications, the data and APIs should be licensed accordingly.

### From vision to reality

- » Support the definition of an open API
- » Use open licences
- » Favour open tools

## Facilitating discovery and use

Cities should aim at ensuring that open APIs are easy to discover and that getting started with them is as easy as possible.

## Building on API standards and technical guidelines

The open APIs should support the established national and international standards for technology and data models in order to ensure their quality. Relationships should be established with standardisation organisations to make sure that the harmonised APIs and standards remain interoperable despite city-specific extensions.

In addition, high-quality and up-to-date documentation and other necessary instructions are to be prepared for the open solutions. The documentation and instructions are then distributed openly for everyone to use.

## Ensuring API-related competence

To ensure that the quality of open APIs is taken into account in the implementations, cities should make certain that the person placing the procurement order has the required competence to include all the necessary information in the invitation to tender and the agreement. Cities should strengthen their open API-related competence by organising suitable training within their organisations and distributing this competence openly to others, where possible. Detailed guidelines can also help accomplish this goal.

### From vision to reality

- » Use open licences
- » Favour open tools
- » Harmonise the data format, data model, or API
- » Create instructions and distribute them openly

### From vision to reality

- » Harmonise the data format, data model, or API
- » Create instructions and distribute them openly

### From vision to reality

- » Create instructions and distribute them openly
- » Arrange training related to open APIs

### **From vision to reality**

- » Aim for a sufficient SLA
- » Support the definition of an open API

## **Making open APIs a part of cities' strategic policies**

Cities should incorporate open APIs into their strategic policies and take open APIs into account in their architecture. As a general rule, open APIs should always be included when procuring new IT systems.

The life cycle of an open API should be designed in the same way as the life cycles of other IT solutions. However, it should be noted that the life cycle of an open API may differ from that of its back-end system.

## **Developing management models for open API standards and products**

A management model to be used inside each city's organisation should be designed for each open API. A joint management model for all cities should be the ultimate goal.

The management model is taken into account in the early stages of planning, whether the open API is going to be used by an individual city or several cities. The design should favour agile development, but also take into account the established principles employed by the cities, for example regarding supported platforms. Any insights made about the API are recorded so that they can be considered when updating each city's own principles.

### **From vision to reality**

- » Create a management model for the APIs

## Strengthening the collaboration between local, national and international actors

Cities benefit from developing their digital interoperability together. They should cooperate to create harmonised API definitions, shared principles and other operating models related to digital interoperability.

In Finland, the Six Cities also cooperate e.g. with the state, the Association of Finnish Local and Regional Authorities, international city networks, local and global developer communities, and other bodies that develop and utilise open APIs and standards.

In addition, it is important to implement standards and solutions that are commonly used on an international level, and cities should also put effort into distributing solutions abroad to make them more commonplace and increase their use potential.

### From vision to reality

- » Create instructions and distribute them openly
- » Aim for a sufficient SLA

# City-specific measures



## Support the definition of an open API

A joint definition of an open API is needed to make sure that APIs are easier to discuss and to better spread awareness of their benefits. In a nutshell, an open API can be defined as an application programming interface whose properties and features are all public and that can be used without any restricting terms and conditions.

However, this does not mean that just anyone can access all the data or the background system behind the API. The open API merely enables easy access to data that has been made available for public distribution. An API can be considered open even if there are terms on the use of the data or if the use of the API requires user authentication or verification of user rights.

The properties of an open API are public and it can be used without any restrictions.

The Six Cities support the open API definition produced through crowd sourcing by COSS, Open Knowledge Finland and the API:Suomi community. Where necessary, the cities take part in the development of the definition.

The Six Cities may also develop other jointly agreed definitions, if needed. For instance, it may be necessary to specify a partially open API that can be provided to partners for a more specific purpose (b2b API).

For example, a partially open API allows a specific user to be granted more extensive access rights than other users. It may also be necessary to specify when it is justified to produce an open API and when it is not.

# Open API definition

*This API definition is produced through crowd sourcing by COSS, Open Knowledge Finland and the API:Suomi community.*

An open Application Programming Interface (API), is an API whose all features are public and which can be used without restricting terms and conditions (for example, it is possible to create an application that utilises the API without explicit approval from the API supplier or mandatory licensing fees).<sup>1</sup> This requires that the API description and any related documentation must be openly available and that the API can be freely used to create and test applications. The use of an open API is free of charge. The user does not need to ask permission from the API provider or indicate the intended purpose of usage in advance.

**Comment:** *No fee will be charged for the API documentation and test data. A fee may be charged for access to the actual information content behind the interface even if the API is open.*

## Application programming interface (API)

An API determines how the software provides information or services to the applications or other information systems.

**Comment:** *The APIs that are of interest from the perspective of this document are usually Web Service interfaces that are used over the Internet. However, the same principles apply to other interface implementations, as well. The open API definition is independent from the technological implementation methods.*

An API can be a simple **data API** through which the data contained by a service can be read to other systems, or it can be a **functional API**, which offers also algorithms or the possibility to change the data in the service via API. If the system has a multiple APIs, it should be specified which of them are open.

---

<sup>1</sup> [www.kdk.fi/fi/kokonaisarkkitehtuuri/sanasto](http://www.kdk.fi/fi/kokonaisarkkitehtuuri/sanasto)

**Comment:** An example of a data API is the interface of the [kansalaisaloite.fi](http://kansalaisaloite.fi) online service<sup>2</sup>, which provides information on citizens' initiatives. Examples of functional APIs are the Helsinki Region Transport's Journey Planner interface<sup>3</sup>, which provides a routing algorithm, and the international Open311 API standard<sup>4</sup> which enables fault reports to be submitted to the compatible city feedback systems.

## Open API

*Relationship: holder-user of the interface*

For an API to be called open it must meet the following conditions:

**1. Openly documented:** The API has been specified and the documentation is openly available and accessible over the Internet. The information contained in the system, its structure and the interfaces have been documented at a sufficient level of detail to ensure that the API is as easy as possible to utilise.

**Comment:** *The documentation must be sufficient for independent development without the user having to request additional information from the provider. If the system utilises non-standard data formats, their structure and processing must also be openly and comprehensively documented.*

**2. Possible to take into use:** The open API can be taken into use without measures from an administrator or system supplier even outside business hours. Possible registrations are handled automatically.

**Comment:** *This does not need to entail access to the production system and therefore does not preclude access rights management.*

**3. Testable:** The API must be testable. At minimum, sample data must be provided for testing purposes. The testability can be implemented in the following ways:

1. open access to the production system, through which service integration is possible, or

---

2 [www.kansalaisaloite.fi/api](http://www.kansalaisaloite.fi/api)

3 [developer.reittiopas.fi/pages/fi/reittiopas-api.php](http://developer.reittiopas.fi/pages/fi/reittiopas-api.php)

4 [www.open311.org](http://www.open311.org)

2. open access to the test system (sandbox), which contains realistic or authentic data, or
3. the test system can be freely downloaded for installation and any use seen fit

The data available through the open API does not need to be open data<sup>5</sup>. The API can be open even if the production system connected to it is entirely isolated from the Internet and can only be accessed by a very limited group of people. If the interface is open but access to the data content is restricted, a test environment that can be accessed openly online must be available.

**Comment:** *The fact that an open API is available for the system does not mean that anyone can access the production system or the information contained by it. For example, a patient information system can involve an open API, but the patient records are not openly accessible. Moreover, an open API can be used to provide information on a particular individual for third parties with individual's consent (MyData).*

The openness of the APIs enables any party to produce a competing software that implements the same interfaces and is, therefore, compatible with all applications that utilise the API.

## API managed by the client

*Relationship: supplier–client*

A client-managed API is an interface that the client is entitled to use and distribute in the way it sees fit. This means that the client can, if it so desires, open the API regardless of the system supplier's stand on the matter. If this is not the case, the API will not be open to external parties and cannot be considered to be an open API.

---

5 [opendefinition.org](http://opendefinition.org)

## Create a management model for the APIs

The implementation of an open API will always affect the internal work processes of a city. Its maintenance must be well-planned in order for the API to benefit the city in the best possible way and for external users to consider the API stable and interesting. A management model is needed for this purpose.

Providing open APIs is a part of the city's services.

Each city should plan an internal management model for the open API at an early stage. The maintenance roles and responsibilities should be clearly determined, as providing open APIs is a part of the city's services. It should be noted that ownership cannot be outsourced. The internal management model of a city must take into account at least the following matters:

- Who owns the service?
- Who is responsible for content maintenance?
- Who is responsible for content development?
- Who is responsible for technical maintenance?
- Who is responsible for technical development?
- Who is responsible for service communication?

When cities are harmonising their APIs, a management model to be used between the cities is also required. The overall aim is to establish a joint open API management model for the Six Cities.

The planning of a joint intercity management model should begin at an early stage in cases where the open API is going to be implemented in two or more cities. Even if each city maintains its own harmonised API, the following matters must be agreed upon together:

- Who owns the API definitions?
- How will the development and maintenance be funded?
- How will the development be coordinated?
- How will the API be developed in practice?
- Who is responsible for version management?

What should be taken into consideration when APIs are replicated between cities?

The possibility of implementing a joint management system should also be explored, as this would enable all harmonised APIs to be accessed from a single location with a single login.

## Use open licences wherever possible

Open licences should be used for open data and open APIs wherever possible. An already established licence should be selected. The terms of data use and the terms of API use as well as the API's source code are licensed separately. The licences must permit the diverse utilisation of the data and the interface.

The terms of data use should be licensed according to the Creative Commons licence family, preferably licence type CC4.0.

The open API's terms of use take into account easy implementation from the user's standpoint. They also define the conditions for the usage and the implementation of the API in each individual system.

The API's source code should be openly licensed, unless something prevents this.

It should be noted that various combinations of licences are possible: for example, it is possible for the API to be open but restrictions to be imposed on the data available through it. However, the aim is to ensure that all licences are open and permit the diverse utilisation of the information. If anything prohibits openness, it must be recorded with justifications.

### Licence

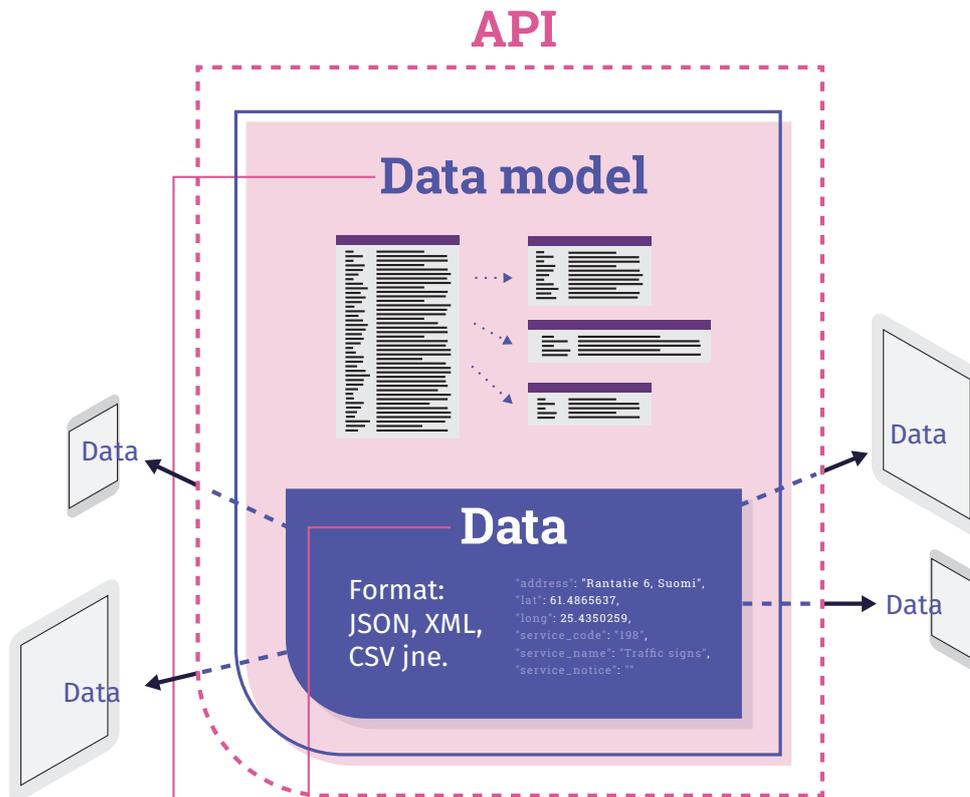
- » the terms of data use
- » the terms of API use
- » the API's source code

## Harmonise the data format, data model or API

It is easier to scale services and expand markets if the open APIs and data formats are harmonised between the cities. In order to further this goal, the cities should harmonise the data format, the API data model and the API itself (see the figure on page 20).

It is advisable to publish the harmonised API even if it is not immediately possible to create a unified data model. If some data cannot be retrieved from the background system, these fields can be left empty. The harmo-

It is advisable to harmonise the data format, the data model and the API, when possible.



● **Data** consists of zeros and ones.  
**Format** is the technical representation of the data.

● **Data model** is the definition of how the data is structured. It also describes what data fields the API contains and how they relate to one another.

nisation of the data model can continue even after the publication of the open API. The most important goal is to ensure that data can be published in as unified a manner as possible. All jointly implemented harmonisations should be documented comprehensively and made widely available, when possible.

## **Aim for a sufficient SLA**

The SLA for an open API is based on a specific API and its specific needs. The aim of the SLA is a sufficient service level for the API, so that users inside and outside the organisation can rely on its operability. In case of an error, the users need to have access to a version of the data that is as up-to-date as possible, even under extraordinary circumstances.

The life cycle of an open API should be designed in the same way as the life cycles of other systems. In the designing process, one should take into account the entire life cycle of the service. When the API's background system is nearing the end of its life, plans must be made on how to replace it or carry out a controlled ramp down of the API in a way that inflicts minimal harm to other systems and external parties.

## **Favour open tools**

Wherever possible, open services should be used for the technical specification of the API and the API's source code version control. A commonly-used open specification language is usually used for the technical specification of the API.

Open source solutions must be used for open API management in order to avoid vendor lock-in. However, the management service itself can be outsourced to an external service provider, as long as it has been established how the management service can be transferred to another service provider, if necessary.

## **Create instructions and distribute them openly**

Creating high-quality, up-to-date instructions and documentations promotes API-related competence and the distribution of open APIs. This requires division of responsibilities and sufficient resourcing.

When procuring an API, the invitation to tender must specify that the documentation is included in the order and indicate how detailed it should be.

It should be noted that instructions need to be created for a variety of parties to ensure the availability of open APIs:

- Technical documentation for software developers (including examples of code and other materials that facilitate use)
- Process descriptions and documentation regarding additional necessary information for the party implementing the system
- Procurement instructions and possible document template for the party making the procurement order
- Other personnel must be provided with general guidelines that specify how the open API affects their day-to-day activities.

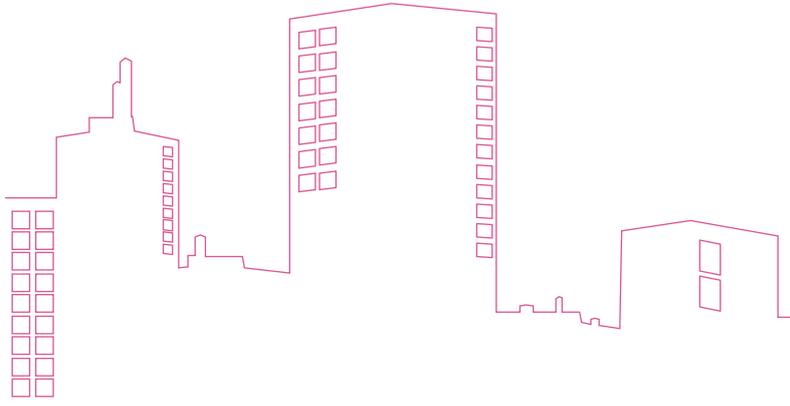
The instructions are to be distributed openly, where this is appropriate.

## **Arrange training related to open APIs**

Staff training should be organised to ensure that the API can be utilised efficiently and that it supplements the existing operating models. The staff should also be able to give support to developers who show interest in the API. For example, the people who process feedback must be trained to handle and respond to feedback submitted through the new API. In addition to this, people responsible for system orders need training on how to monitor the quality of the ordered API.

# Checklist for opening an open API

1. Publish public information as open data, unless there is a specific reason to do otherwise.
2. Open up the data in a machine-readable format and under terms that allow reuse.
3. Strive to utilise open, standardised and harmonised APIs.
4. Utilise internationally established standards.
5. Use the API in your city organisation's own operations.
6. Document the API well to enable utilisation outside your organisation.
7. Design a life cycle and a management model for the API.
8. Ensure a sufficient SLA.



# Contact Us

## **6Aika API cooperation**

Forum Virium Helsinki  
*info@forumvirium.fi*

### **Helsinki**

*hri@hel.fi*

### **Espoo**

*ict-palvelut@espoo.fi*

### **Vantaa**

*ictpalvelut@vantaa.fi*

### **Tampere**

*avoindata@tampere.fi*

### **Turku**

*avoindata@turku.fi*

### **Oulu**

*avoindata@ouka.fi*