

# Versioning

Memi Lavi  
[www.memilavi.com](http://www.memilavi.com)



# Importance of Versioning

---

- Your API will be updated
- You don't control your clients
- You can't force version upgrade

# Importance of Versioning

- Set version support policy

## API End-of-Life

Salesforce is committed to supporting each API version for a minimum of **three years** from the date of first release. In order to mature and improve the quality and performance of the API, versions that are more than three years old might cease to be supported.

When an API version is to be deprecated, advance notice is given at least one year before support ends. Salesforce will directly notify customers using API versions planned for deprecation.

[https://developer.salesforce.com/docs/atlas.en-us.api\\_meta.meta/api\\_meta/meta\\_support\\_policy.htm](https://developer.salesforce.com/docs/atlas.en-us.api_meta.meta/api_meta/meta_support_policy.htm)

# Importance of Versioning

---

- Include versioning in the API

# Versioning In The API

- Version can come in 3 forms:
  - In the URL

```
1 | curl https://yourInstance.salesforce.com/services/data/v20.0/query/?q=SELECT+name+from+Account
```

[https://developer.salesforce.com/docs/atlas.en-us.api\\_rest.meta/api\\_rest/dome\\_query.htm](https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/dome_query.htm)

# Versioning In The API

---

- Version can come in 3 forms:
  - In a Header

```
GET /users/3 HTTP/1.1
```

```
Accept: application/vnd.myname.v1+json
```

# Versioning In The API

---

- Version can come in 3 forms:
  - In a Query Parameter

```
http://example.com/users/1234?version=4
```

# What To Use?

---

- Header is the most correct form
  - Adheres to the REST principles
- URL is the most common and easiest to implement
- DO NOT use Query Params