## Prevention

While prepared statements/stored procedures can prevent injections in SQL queries, not all programming languages and libraries provide an equivalent for XPath queries. Therefore, proper (manual) sanitization is the only universal method of preventing XPath injection vulnerabilities.

Generally, we must treat all user input as untrusted and perform sanitization before inserting it into an XPath query. The simplest and most secure way is implementing a whitelist that only allows alphanumeric characters in the user input inserted into the XPath query. The web application can then reject any input that contains characters that are not whitelisted.

Additionally, verifying the expected data type and format when performing sanitization is crucial. If the web application expects an integer, it must verify that the user input consists of only digits. When applicable, we can additionally perform checks for semantical correctness. For instance, if a variable can only assume a fixed set of values, we can check that the user input conforms to these semantical rules in addition to the syntactical ones. An example would be the GET parameter f in the previous sections, which can only assume the values fullstreetname and streetname. The web application can thus check if the user input matches one of these values and is thus semantically correct.

Alternatively to the whitelist approach, a blacklist approach blocking the following XPath control characters is also sufficient, though a whitelist is always preferable:

* Single quote: '
* Double quote: "
* Slash: /
* At: @
* Equals: =
* Wildcard: \*
* Brackets: [, ], and parentheses (, )