



WORKING WITH WEB SERVICES

WORKING WITH WEB SERVICES

- Examples of Web Services
 - Check the price of an item
 - Place an order for a product
 - Update contact information
 - Check the weather
- A Web Service exchanges information between applications using the standard Web protocol
 - Usually the data is in the form of JSON or XML
- Web requests have two main components
 - **Header** - Describes the message
 - The header consists of simple concise text lines of Keyword: Value and primarily defines content size, type, source and destination
 - **Body** - Contains the message

WORKING WITH WEB SERVICES

- The processing of Web requests begins with a URL
 - `http://target-system-name/target...?parameters`
- Two basic types of requests
 - **GET** - Requests the contents of a target from the server
 - **POST** - Sends some data (i.e. a form) to a specific target on the server

WORKING WITH WEB SERVICES

- PxPlus provides a utility for submitting a request with data and returning the result
"*plus/web/request"
- Supports secure (**https://**) and unsecure (**http://**) requests
- Uses HTTP version 1.1 protocol
 - Can force HTTP version 1.0 protocol by defining the global variable **%web_request_http_ver\$** as "1.0"
- By default, the user agent will be the same as if the request came from a Windows version of Google Chrome with the version of PxPlus appended to it
 - Can be overridden by including a user agent header in the **extrahdrs\$** argument
extrahdr\$="User-Agent: MyApp/1.00.0000"
- Default request is a **GET** if no post data is provided; otherwise, a **POST** is used
- Eight entry points for additional request types are available: **GET, POST, DELETE, PUT, TRACE, OPTIONS, PATCH** and **HEAD**
- This utility also allows you to specify the content type, SSL certificate and additional headers

WORKING WITH WEB SERVICES

CALL `*plus/web/request`, url\$, postdata\$, recvdata\$, recvhdr\$, mimetype\$, certificate\$, extrahdrs\$, method\$, timeout

Argument	Description
url\$	Specifies target server and if secure or non-secure
postdata\$	Data to send to the server
recvdata\$	String variable to receive response
recvhdr\$	Header of response received
mimetype\$	Optional content type of data
certificate\$	Optional path of certificate, if needed
extrahdrs\$	Optional additional headers to include on send
method\$	HTTP request method
timeout	Timeout value (if not provided, defaults to 15 seconds)

WORKING WITH WEB SERVICES

Example - Web request getting current weather for Toronto

```
CALL "*plus/web/request","https://goweather.herokuapp.com/weather/toronto","", Resp$,Hdr$,"","", "",30  
dim load forecast${all}=resp$  
msgbox "It is: "+forecast$["temperature"],"Weather"
```

