

# Sheilta KeepAlive Ext.

Enough worrying about refreshing

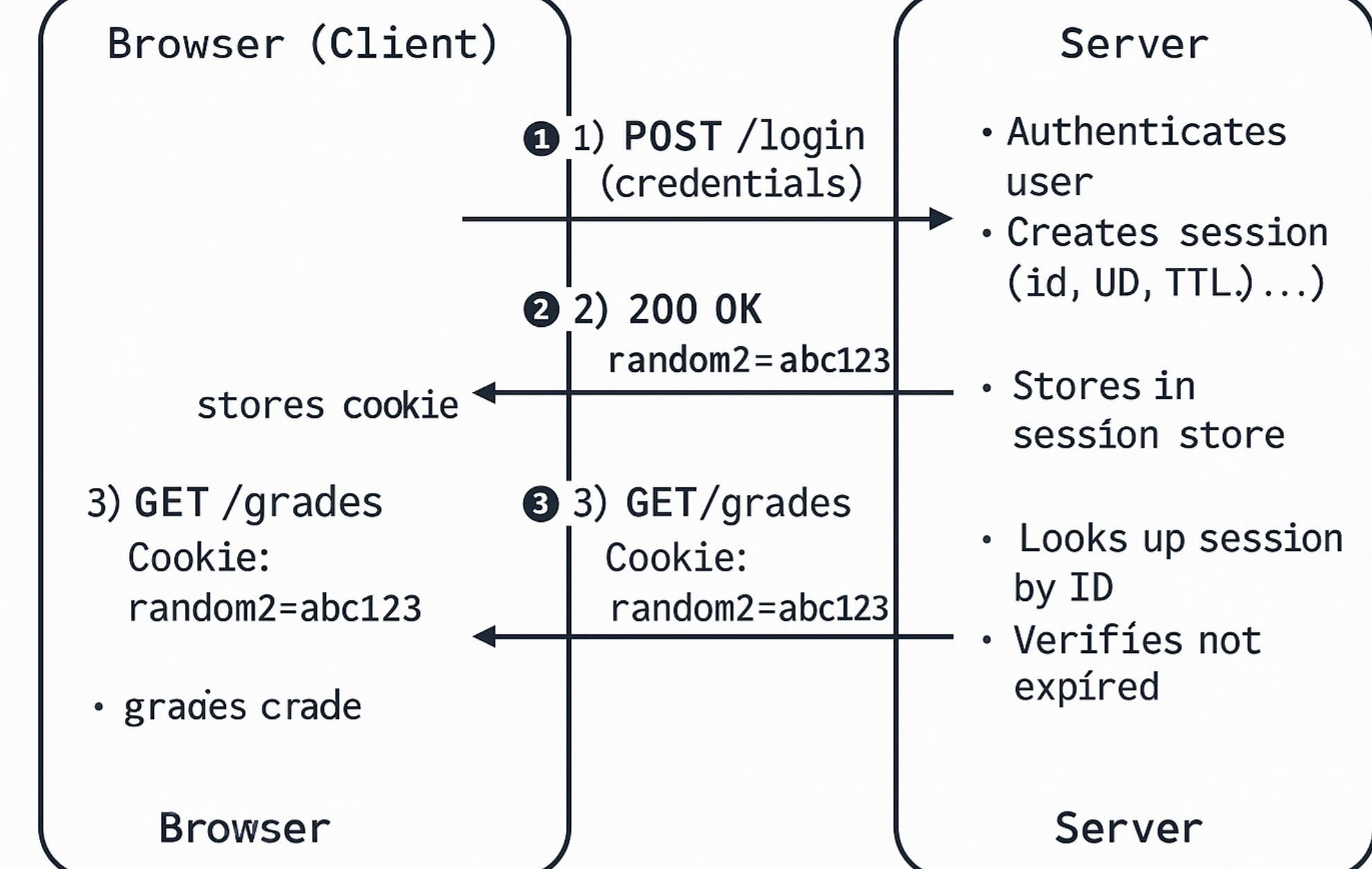
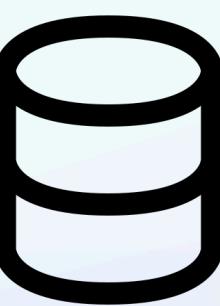
# Introduction

What even are cookies? Session cookies?

- Small Key-Value pairs stored by browser
- Sent automatically with every HTTP/S request to same domain

## • Session Cookies

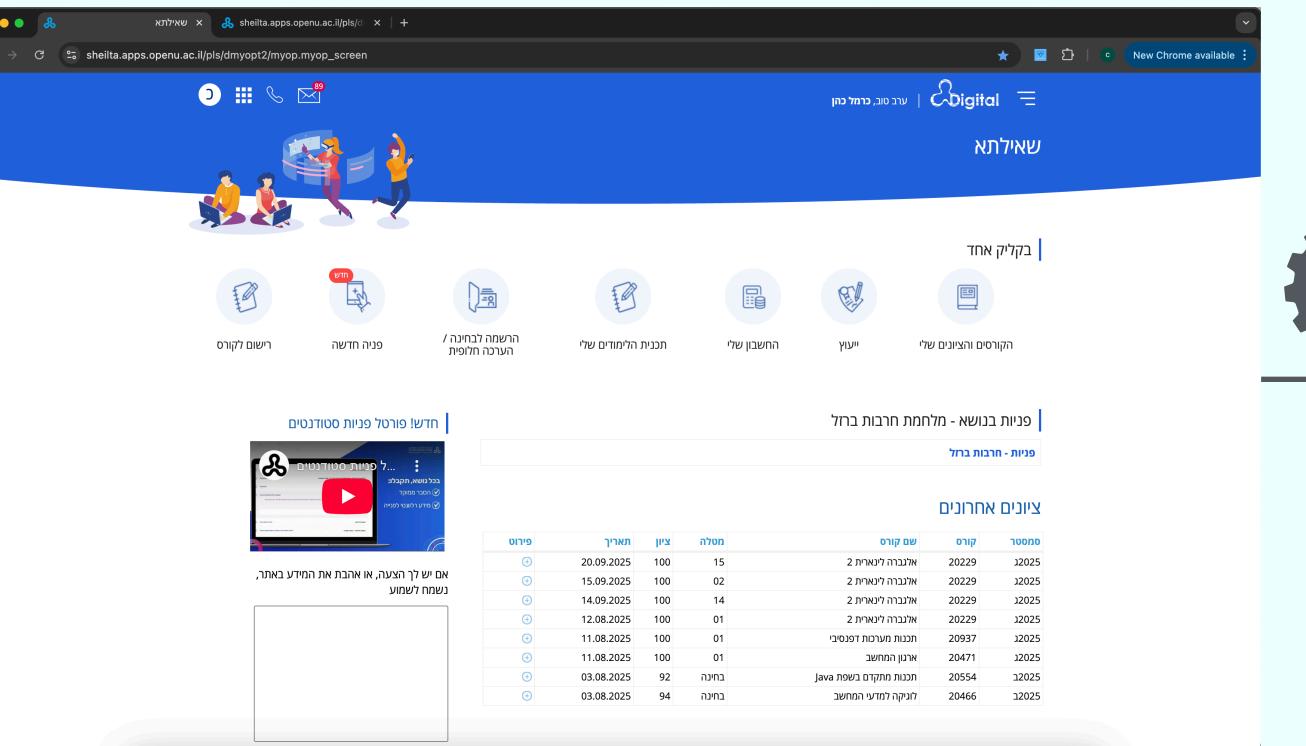
- Server uses the ID to map user → session record
- Authenticate the user on each request
- Save the user from re-logging in every time



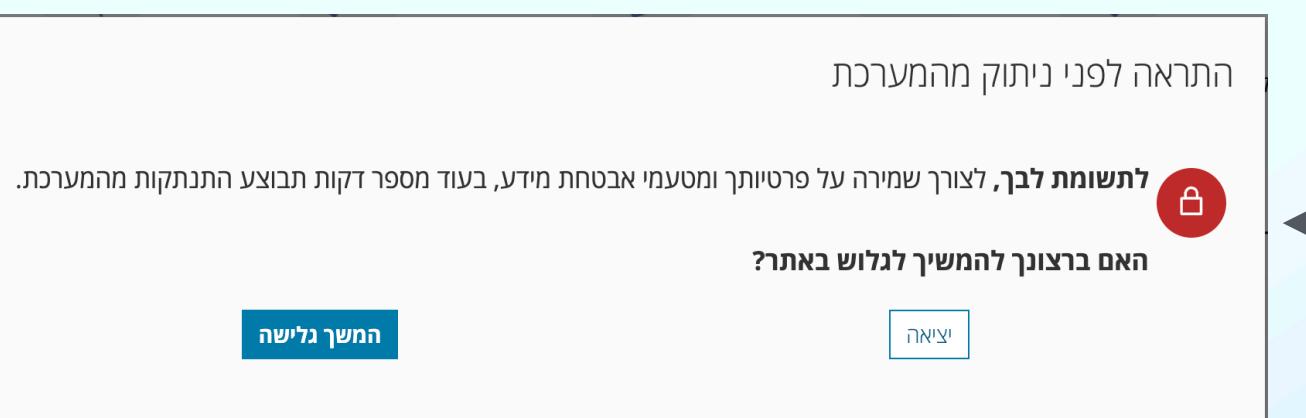
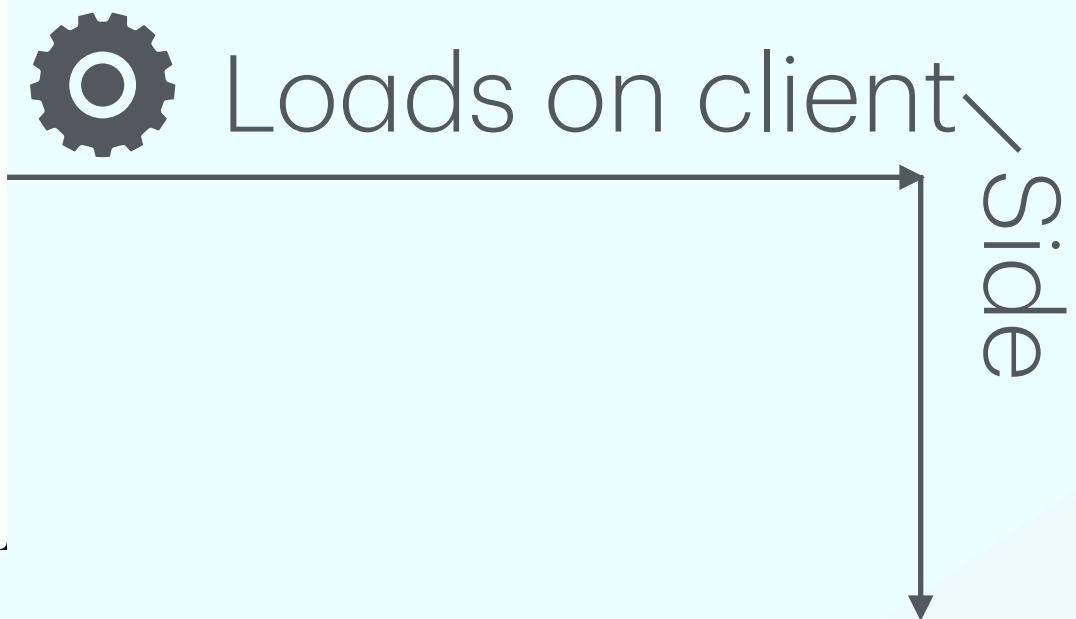
# The Problem

30 minutes is not enough...

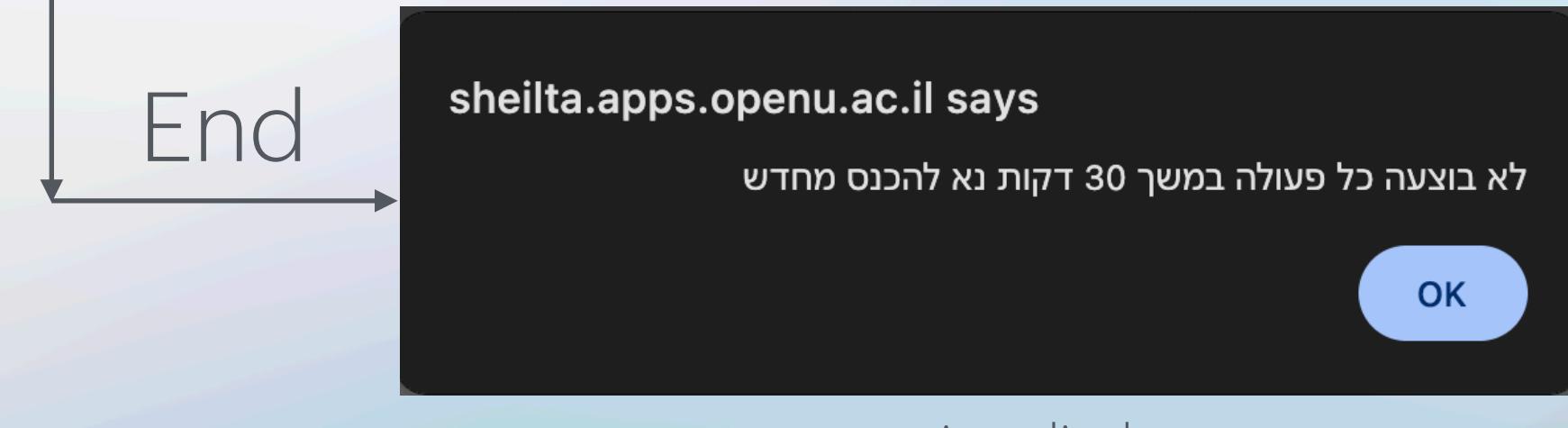
- Sheilta logs out after 30 minutes of inactivity
- Frustrating when watching videos on another domain → Sheilta tabs expire
- **Motivation:** automate session keeping without manual refresh



sheilta main page



~10 sec warning



session died

```
// version 1.0.7 20/11/2010
// version 1.0.6 34/09/2010
// version 1.0.5 30/06/2010

//checkUrl = https://sheilta.apps.openu.ac.il/pls/dmyopt2/session_guard
//refreshUrl = https://sheilta.apps.openu.ac.il/pls/dmyopt2/session_guard?p_refresh=1
//logoutUrl = https://www.openu.ac.il/students/pages/default.aspx/logout=1

function SessionGuard() {
    //<summary>
    //</summary>
    //<param name="checkUrl" type="String">
    //<param name="refreshUrl" type="String">
    //<param name="logoutUrl" type="String">
    //</summary>

    // Required settings
    this.checkUrl = null;
    this.refreshUrl = null;
    this.logoutUrl = null;

    this.defaultInterval = 30;
    this.refreshInterval = 10;
    this.minimumInterval = 10;
    this.dialogId = "session-guard";
    this.dialogYesId = this.dialogId + "-yes";
    this.dialogNoId = this.dialogId + "-no";

    var _this = this;
    var idDialogRaised = false;
    var logoutTimeoutId;

    this.start = function () {
        //<summary>
        //</summary>
        // Start the session guard timer.
        // The properties checkUrl, refreshUrl and logoutUrl must be set before this call.
        if (!_this.checkUrl) {
            console.error("SessionGuard error: checkUrl is required.");
            return;
        }

        if (!_this.refreshUrl) {
            console.error("SessionGuard error: refreshUrl is required.");
            return;
        }

        if (!_this.logoutUrl) {
            console.error("SessionGuard error: logoutUrl is required.");
            return;
        }

        setTimeout(check, _this.defaultInterval * 1000);
    };

    this.initial = function () {
        //<summary>
        //</summary>
        //<param name="checkUrl" type="String">
        //<param name="callback" type="Function">
        this.start();
    };

    var check = function () {
        requestUrl(_this.checkUrl, checkCallback);
    };

    var checkCallback = function (nextIntervalText) {
        var nextInterval = nextIntervalText ? new Number(nextIntervalText) : null;

        if (isNaN(nextInterval)) {
            setTimeout(check, _this.defaultInterval * 1000);
            return;
        }

        if (nextInterval < _this.minimumInterval) {
            setTimeout(check, _this.defaultInterval * 1000);
            return;
        }

        if (nextInterval > _this.refreshInterval) {
            _this.refresh();
        }
    };
}
```

session-guard-1.0.js

# Investigation Process

Curiosity kicks in..

- **Explored login/logout flow**

- **Login** → session-guard-1.0.js loaded
- **Logout** → different APIs for Sheilta vs course tabs

- **Session-guard-1.0.js revealed**

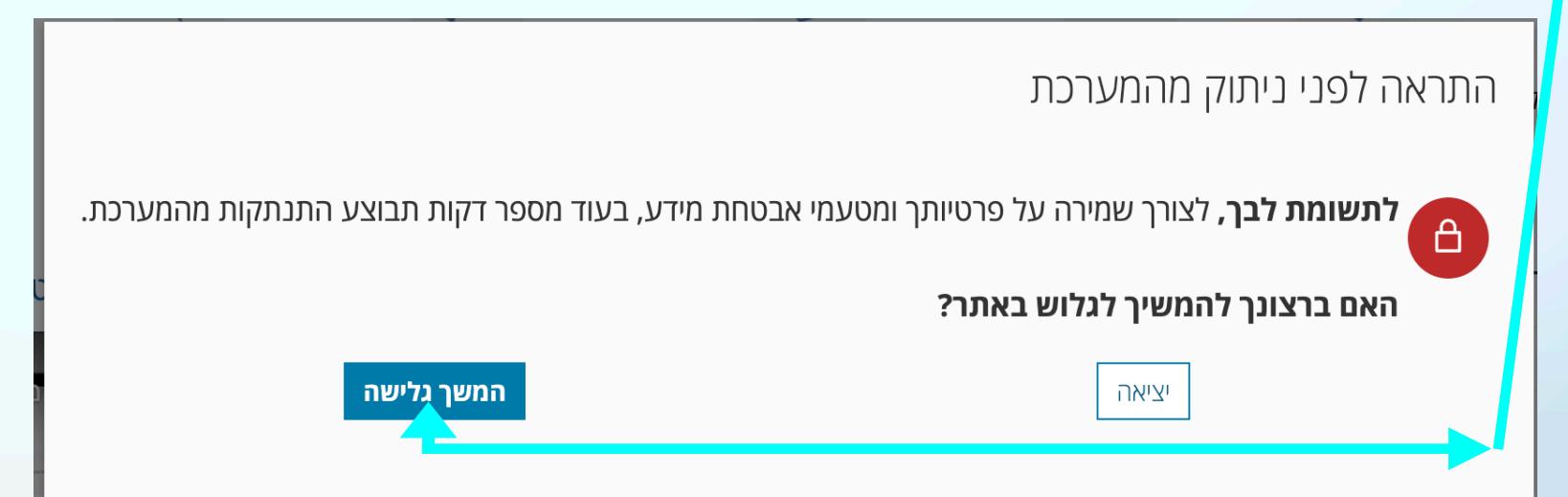
- **checkUrl** → returns session TTL (seconds)
- **refreshUrl** → resets session cookie
- **logoutUrl** → forces logout

```
<script type="text/javascript" src="/ows-img/_default/scripts/session-guard-1.0.js"></script>
<script type="text/javascript">
    var sao = new SessionGuard();
    sao.checkUrl = "session_guard";
    sao.refreshUrl = "session_guard?p_refresh=1";
    sao.logoutUrl = "https://www.openu.ac.il/students/pages/default.aspx?logout=1";
    sao.initial();
</script>
```

```
var refresh = function () {
    if (logoutTimeoutId) {
        clearTimeout(logoutTimeoutId);
        logoutTimeoutId = false;
    }

    requestUrl(_this.refreshUrl, checkCallback);
};

var logout = function () {
    if (_this.logoutUrl) {
        top.window.location.href = _this.logoutUrl;
    }
};
```

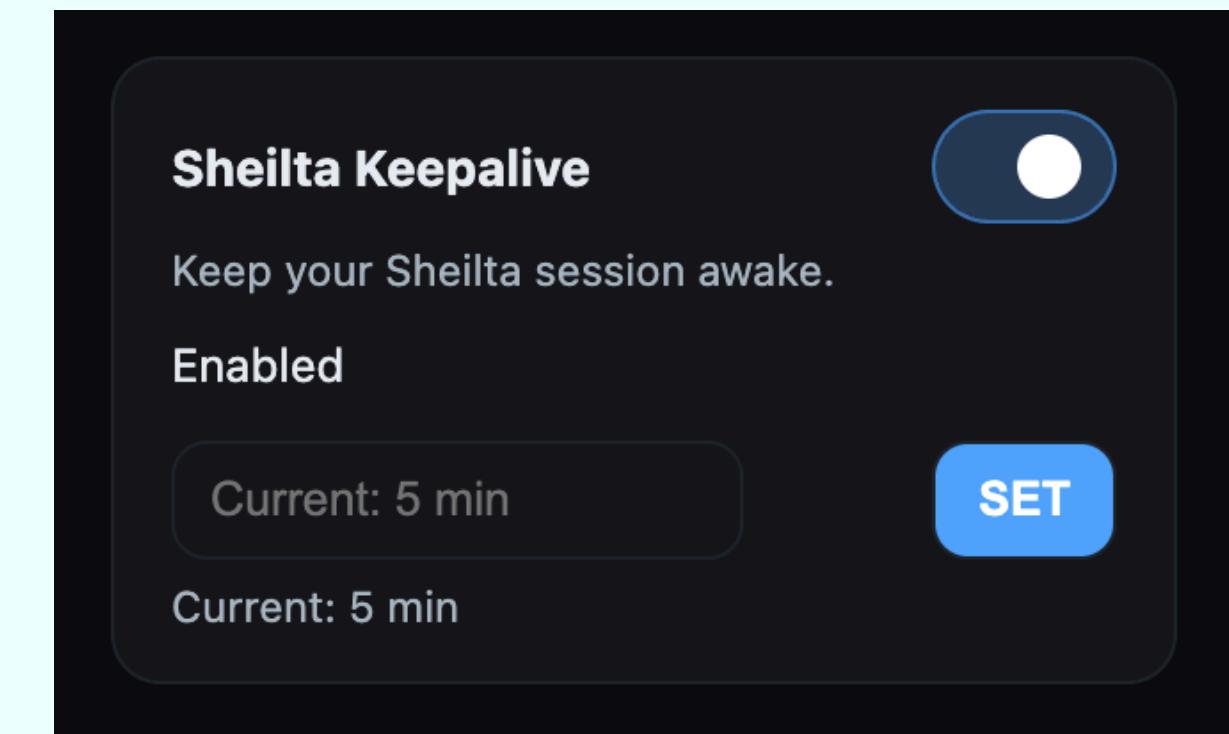


```
function requestUrl(url, callback) {
    if ($) {
        $.ajax(url, {
            method: "POST",
            success: callback,
            error: function (error) {
                callback(null);
            },
            xhrFields: { withCredentials: true },
        });
    } else {
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.withCredentials = true;
        xmlhttp.onreadystatechange = function () {
            if (xmlhttp.readyState == 4) {
                if (xmlhttp.status == 200) {
                    callback(xmlhttp.responseText);
                } else {
                    callback(null);
                }
            }
        };
        xmlhttp.open("POST", url, true);
        xmlhttp.send();
    }
}
```

# Solution Design

Chrome loves extensions...

- Runs quietly in background
- Periodically calls `refreshUrl`
- Key features:
  - Toggleable On/Off
  - Configureable interval
  - Smart check → only if Sheilta tab open



Name	Status	Type	Initiator	Size	Time
popup.html	200	document	Other	0.4 kB	3 ms
popup.js	200	script	popup.html:17	0.8 kB	2 ms
session_guard?p_refresh=1	200	fetch	background.js:57	0.3 kB	417 ms

```
11:13:37.000 [KeepAlive] Alarm @ 5 min
11:13:40.259 [KeepAlive] Alarm cleared
11:13:40.943 [KeepAlive] Alarm @ 5 min
11:13:43.517 [KeepAlive] Alarm @ 10 min
11:13:45.194 [KeepAlive] Alarm @ 2 min
11:13:46.253 [KeepAlive] Alarm cleared
11:13:46.756 [KeepAlive] Alarm @ 2 min
```

# Code Walkthrough

Enough with the talking...

- **Alarm setup** → triggers every X minutes
- **Tab detection** → verifies Sheilta tab exists
- **Keepalive fetch** → calls refreshUrl with cookies

```
const ALARM_NAME = "sheiltaKeepAlive";
const SHEILTA_REFRESH_URL =
  "https://sheilta.apps.openu.ac.il/pls/dmyopt2/session_guard?p_refresh=1";

let enabled = true;
let intervalMin = 5;

chrome.storage.local.get(
  { sheiltaEnabled: enabled, sheiltaIntervalMin: intervalMin },
  ({ sheiltaEnabled, sheiltaIntervalMin }) => {
    enabled = !sheiltaEnabled;
    intervalMin = Number(sheiltaIntervalMin) || intervalMin;
    applyEnabledState(enabled);
  }
);
```

```
function startAlarm() {
  chrome.alarms.clear(ALARM_NAME, () => {
    chrome.alarms.create(ALARM_NAME, { periodInMinutes: intervalMin });
    console.log("[KeepAlive] Alarm @", intervalMin, "min");
  });
  function stopAlarm(): void
}

function stopAlarm() {
  chrome.alarms.clear(ALARM_NAME, () => {
    console.log("[KeepAlive] Alarm cleared");
  });
}

chrome.runtime.onInstalled.addListener(() => applyEnabledState(enabled));
chrome.runtime.onStartup.addListener(() => applyEnabledState(enabled));
```

```
function hasSheiltaTab(cb) {
  chrome.tabs.query({}, (tabs) => {
    cb(
      tabs.some(t =>
        /^https:\/\/sheilta\.apps\.openu\.ac\.il/i.test(t.url || "")
      )
    );
  });
}
```

```
chrome.alarms.onAlarm.addListener(alarm => {
  if (alarm.name !== ALARM_NAME) return;
  if (!enabled) {
    console.log("[KeepAlive] Disabled. Skip.");
    return;
  }

  hasSheiltaTab(has => {
    if (!has) {
      console.log("[KeepAlive] No Sheilta tab. Skip.");
      return;
    }

    fetch(SHEILTA_REFRESH_URL, { method: "GET", credentials: "include" })
      .then(res => {
        if (res.ok) console.log("[KeepAlive] refresh OK:", res.status);
        else console.warn("[KeepAlive] refresh non-OK:", res.status);
      })
      .catch(err => console.error("[KeepAlive] refresh error:", err));
  });
});
```

# LET'S DEMO THIS

[https://sheilta.apps.openu.ac.il/pls/dmyopt2/myop.myop\\_screen](https://sheilta.apps.openu.ac.il/pls/dmyopt2/myop.myop_screen)