

API Based Approach (REST API)

Mechanism: Push Based, external system needs to consume Darwinbox APIs to push the attendance punches.

Frequency: Recommendation will be given as per the employee count and expected punches (e.g., API can be called every 15 mins, running a batch process of 100 punches per payload).

Procedure:

STEP 1. (Setting Configurations)

Darwinbox will host a public endpoint URL, that needs to be consumed by Clients/Biometric SPOC to push the attendance Punches. Here, multiple attendance punches can be pushed by creating multi array of employee IDs. (Recommendation will be given at the time of implementation)

STEP 2. (Testing on UAT)

If required Client/Biometric SPOC can directly use the API creds of the production instance to test the punches, else Integrations Team can help with a sandbox environment (other testing instance and not client's staging instance) to test the same.

STEP 3. (Production Movement)

Post confirmation the Integration can be made live for all employee in the Production instance.

Sample API Endpoint URL:

Method	URL
POST	https://<clientsubdomain>.darwinbox.in/AttendancePunchesApi

Authentication

Basic Authentication: Basic auth username and password will be shared by Darwinbox

Parameters:

Parameters	Value/Description	Mandatory/Non-Mandatory
api_key	Darwinbox API Key for authorization	Mandatory
id	Unique ID of the punch (This is not mandatory)	Non-Mandatory
timestamp	Timestamp of the Attendance Punch (DD-MM-YYYY HH:MM:SS)	Mandatory
machineid	Unique ID of the machine	Mandatory (if this value is not stored/any hardcoded value to be passed)
status	Directionality of the punch if available, 1 = IN, 2 = OUT	**Non-Mandatory

** Mandatory only if directionality has been enabled and if functional aspects like work duration and break duration, etc. are to be shown in Darwinbox.

API Status Code

Status	Response
200	Attendance Punches Recorded Successfully
400	Attendance Input Parameter Not Found
401	Invalid Credentials
500	Internal Server Error

Sample Payload:

```
{
  "api_key": "56a4c2b14efa17e8e4fb9b6df1bd5517280cf978cc83d7365b9bd7297281458ab6a
a43dce64b48c4208751234567890asdfghjkertcvb14f40b7b3",
  "attendance": {
    "EMP001": [
      {
        "id": "1642511891684",
        "timestamp": "2022-03-08 07:48:11",
        "machineid": "HA13YABK",
        "status": 1
      },
      {
        "id": "1642511891685",
        "timestamp": "2022-03-08 17:40:19",
        "machineid": "HA14YABK",
        "status": 2
      },
      {
        "id": "1642511891686",
        "timestamp": "2022-03-08 18:40:30",
        "machineid": "HA14YABK",
        "status": 1
      }
    ],
    "EMP002": [
      {
        "id": "1642511891680",
        "timestamp": "2022-03-08 07:50:21",
        "machineid": "HA13YABK",
        "status": 1
      },
      {
        "id": "1642511891681",
        "timestamp": "2022-03-08 19:40:19",
        "machineid": "HA14YABK",
        "status": 2
      }
    ]
  }
}
```

Sample Output (Success):

```
{
  "status": 1,
  "message": "Attendance Punches Recorded Successfully",
  "attendance_status": {
    "EMP001": {
      "ref_id": "a624dd45198a1b",
      "status": 1,
      "message": "Saving success"
    },
    "EMP002": {
      "ref_id": "a624dd45199117",
      "status": 1,
      "message": "Saving success"
    }
  }
}
```

The biometric vendor/client is expected/recommended to store the reference ID (ref_id) and if possible, the request payload in logs. In case of an issue, this ref_id can be provided to the Darwinbox account management team for validation.

Sample Architecture

