



SwipeLab

API Contract

This document describes the API calls from backend, with detailed requests and responses.

Contents

API Contract	1
Authentication Endpoints	2
1.1 Registration	3
1.2 Login	5
1.3 Logout (Token Revocation)	6
1.4 Refresh Token	6
1.5 Current User	7
1.6 Forgot Password	7
1.7 Reset Password	7
1.8 Email Verification	8
1.8.1 Send Verification To Email	8
1.8.2 Verify Email (Confirm)	8
Classification Endpoints	9
2.1 Get Next Batch	10
2.2 Submit Classification	10
2.3 Progress	11
2.4 User Statistics	11
Leaderboard Endpoints	15
Dashboard Endpoints	16
3.1 View Tasks	17
3.2 Create Task	18
3.3 Archive Task	19
3.4 Update Task	19
3.5 Activate Task	21
3.6 Pause Task	22
3.7 Task Analytics	22
3.8 Export Results	25
3.9 View Gold Images	26
4.0 Add Gold Image	27
4.1 Delete Gold Image	27
4.2 View Recipients List	28
4.3 Create Recipients List	28
4.4 Delete Recipients List	29
4.5 Update Recipients List	29
4.6 View Taxonomy	30
5.1 My Tasks	31
5.2 Task Details	32
5.3 Play Task	33

Authentication Endpoints

Auth API Summary (SwipeLab)

	Purpose	Endpoint
1.1.1	Register	POST /auth/register
1.1.2	Register (Google)	POST /auth/register/google
1.2.1	Login	POST /auth/login
1.2.2	Login (Google)	POST /auth/login/google
1.3	Logout	POST /auth/logout
1.4	Refresh token	POST /auth/refresh
1.5	Current user	GET /auth/me
1.6	Forgot password	POST /auth/password/forgot
1.7	Reset password	POST /auth/password/reset
1.8	Email verification	POST /auth/email/*

1.1 Registration

1.1.1 Via Username & Password:

```
POST /api/v1/auth/register
Request:
{
  "username": "john_doe",
  "email": "user@example.com",
  "password": "P@ssw0rd!"
}

Response:
```

Commented [pw1]: Auto-login after successful registration (very common UX).

1.1.2 Via Google:

```
POST /api/v1/auth/register/google
Request:
{
  "googleToken": "eyJhbGciOiJSUzI1NiIsImtpZCI6..."
}
```

Response:

```
{ "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "expiresIn": 2592000,
  "user": {
    "id": 1,
    "email": "user@example.com",
    "username": "john_doe",
    "role": "USER"
  }
}
```

Commented [pw2]: Auto-login after successful registration (very common UX).

Error Responses:

- **Username already exists**

```
{  
  "errorCode": "USERNAME_ALREADY_EXISTS",  
  "message": "Username is already taken"  
}
```

- **Email already exists**

```
{  
  "errorCode": "EMAIL_ALREADY_EXISTS",  
  "message": "Email is already registered"  
}
```

- **Email not allowed**

```
{  
  "errorCode": "EMAIL_NOT_ALLOWED",  
  "message": "Email is not allowed"  
}
```

- **Weak password**

```
{  
  "errorCode": "WEAK_PASSWORD",  
  "message": "Password does not meet security requirements"  
}
```

- **Invalid input**

```
{  
  "errorCode": "VALIDATION_ERROR",  
  "message": "Invalid registration data"  
}
```

1.2 Login

Via Username & Password:

```
POST /api/v1/auth/login
Request:
{
  "username": "john_doe",
  "password": "P@ssw0rd!"
}

Response:
{
  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "expiresIn": 2592000,
  "user": {
    "id": 1,
    "email": "user@example.com",
    "username": "john_doe",
    "role": "USER"
  }
}
```

- *Via Google:*

```
POST /api/v1/auth/login/google
Request:
{
  "googleToken": "eyJhbGciOiJSUzI1NiIsImtpZCI6..."
}

Response:
{
  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "expiresIn": 2592000,
  "user": {
    "id": 1,
    "email": "user@example.com",
    "username": "john_doe",
    "role": "USER"
  }
}
```

1.3 Logout (Token Revocation)

```
POST /api/v1/auth/logout
Request:
{
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}

Response:
{
  "message": "Logged out successfully"
}
```

1.4 Refresh Token

```
POST /api/v1/auth/refresh
Request:
{
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}

Response:
{
  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "expiresIn": 2592000
}
```

Commented [v2]: Access token is NOT sent here (it may be expired)

Error Responses:

- Invalid token

```
{
  "errorCode": "INVALID_REFRESH_TOKEN",
  "message": "Refresh token is invalid or expired"
}
```

- Token revoked

```
{
  "errorCode": "REFRESH_TOKEN_REVOKED",
  "message": "Refresh token has been revoked"
}
```

1.5 Current User

```
GET /api/v1/auth/me
Request:
Headers:
  Authorization: Bearer <accessToken>
Body:
  (none)

Response:
{
  "id": 1,
  "email": "user@example.com",
  "username": "john_doe",
  "role": "USER"
}
```

1.6 Forgot Password

```
POST /api/v1/auth/password/forgot
Request:
{
  "email": "user@example.com"
}

Response:
{
  "message": "If the email exists, a reset link has been sent"
}
```

1.7 Reset Password

```
POST /api/v1/auth/password/reset
Request:
{
  "resetToken": "abc123resetToken",
  "newPassword": "NewP@ssw0rd!"
}

Response:
{
  "message": "Password has been reset successfully"
}
```

1.8 Email Verification

1.8.1 Send Verification To Email

```
POST /api/v1/auth/email/verify  
Request:  
Headers:  
  Authorization: Bearer <accessToken>  
Body:  
  (none)
```

Commented [pw4]: USED:
After registration
When user requests resend

```
Response:  
{  
  "message": "Verification email sent successfully"  
}
```

1.8.2 Verify Email (Confirm)

```
POST /api/v1/auth/email/confirm  
Request:  
{  
  "verificationToken": "abc123emailVerificationToken"  
}
```

Commented [pw5]: Triggered when the user clicks the link
in the email.

```
Response:  
{  
  "message": "Password has been reset successfully"  
}
```

Classification Endpoints

	Purpose	Endpoint
2.1	Get next batch	GET /classifications/next-batch
2.2	Submit classification	POST /classifications/{id}/submit
2.3	User progress	GET /classifications/progress
2.4.1	User overview	GET /statistics/me
2.4.2	User vs experts	GET /statistics/me/vs-experts
2.4.3	User vs users	GET /statistics/me/vs-users
2.4.4	Performance breakdown	GET /statistics/me/breakdown
2.4.5	Time series	GET /statistics/me/timeseries

2.1 Get Next Batch

```
GET /api/v1/classifications/next-batch?count=10
Request:
Headers:
Authorization: Bearer <accessToken>
{
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}

Response:
{
  "images": [
    {
      "imageId": 1234,
      "taskId": 5,
      "question": "Is this a wasp?",
      "image": {
        "contentType": "image/jpeg",
        "data": "<base64-encoded-image>"
      },
      "referenceImages": [
        {
          "contentType": "image/jpeg",
          "data": "<base64-encoded-image>",
          "caption": "Example of a wasp - note the narrow waist"
        }
      ]
    }
}
```

2.2 Submit Classification

```
POST /api/v1/classifications/{classificationId}/submit
Request:
Headers:
Authorization: Bearer <accessToken>
{
  "imageId": 1234,
  "taskId": 5,
  "question": "Is this a wasp?",
  "decision": "YES",
  "responseTimeMs": 2340
}
```

Response:

```
{  
  "images": [  
    {  
      "id": 1234,  
      "taskId": 5,  
      "question": "Is this a wasp?",  
      "image": {  
        "contentType": "image/jpeg",  
        "data": "<base64-encoded-image>"  
      }  
    }  
  ],  
  "referenceImages": [  
    {  
      "contentType": "image/jpeg",  
      "data": "<base64-encoded-image>",  
      "caption": "Example of a wasp - note the narrow waist"  
    }  
  ]  
}
```

2.3 Progress

GET /api/v1/classifications/progress

Request:

Headers:

Authorization: Bearer <accessToken>

Body:

(none)

Response:

```
{  
  "completed": 120,  
  "accuracy": 0.91  
}
```

2.4 User Statistics

Base Path:

/api/v1/statistics

All endpoints require authentication.

2.4.1 User Classification Statistics

```
GET /api/v1/statistics/me
Request:
Headers:
Authorization: Bearer <accessToken>
Body:
(none)

Response:
{
  "summary": {
    "totalClassifications": 320,
    "correctClassifications": 295,
    "accuracy": 0.92,
    "contributionPercentage": 0.42, // % of total dataset
    "rank": {
      "daily": 3,
      "weekly": 5,
      "monthly": 12,
      "allTime": 45
    },
    "rankPercentile": 87
  },
  "trend": {
    "byDay": [
      { "date": "2025-12-10", "accuracy": 0.89 },
      { "date": "2025-12-11", "accuracy": 0.91 },
      { "date": "2025-12-12", "accuracy": 0.94 }
    ]
  }
}
```

2.4.2 User Vs Experts

```
GET /api/v1/statistics/me/vs-experts
Request:
Headers:
Authorization: Bearer <accessToken>
Body:
(none)
```

```
Response:  
{  
  "user": {  
    "accuracy": 0.92  
  },  
  "experts": {  
    "accuracy": 0.97  
  },  
  "difference": {  
    "accuracy": -0.05,  
  }  
}
```

2.4.3 User Vs Community Comparison

```
GET /api/v1/statistics/me/vs-users  
Request:  
Headers:  
Authorization: Bearer <accessToken>  
Body:  
  (none)
```

```
Response:  
{  
  "user": {  
    "accuracy": 0.92  
  },  
  "experts": {  
    "accuracy": 0.97  
  },  
  "difference": {  
    "accuracy": -0.05  
  }  
}
```

2.4.4 Detailed Performance Breakdown

```
GET /api/v1/statistics/me/breakdown  
Request:  
Headers:
```

```
Authorization: Bearer <accessToken>
Body:
  (none)
```

Response:

```
{
  "byCategory": [
    {
      "category": "WASP",
      "accuracy": 0.95,
      "total": 120
    },
    {
      "category": "BEE",
      "accuracy": 0.88,
      "total": 200
    }
  ]
}
```

2.4.5 Time-Series Data (Graph-Ready)

```
GET /api/v1/statistics/me/timeseries?metric=accuracy&period=30d
Request:
Headers:
Authorization: Bearer <accessToken>
Body:
  (none)
```

Response:

```
{
  "metric": "accuracy",
  "points": [
    { "timestamp": "2025-12-01", "value": 0.88 },
    { "timestamp": "2025-12-02", "value": 0.90 }
  ]
}
```

Leaderboard Endpoints

```
GET /api/v1/leaderboard/weekly?page=0&size=10
```

Request:

Headers:

```
Authorization: Bearer <accessToken>
```

Body:

```
(none)
```

Response:

```
{
  "period": "weekly",
  "startDate": "2024-12-09",
  "endDate": "2024-12-15",
  "entries": [
    {
      "rank": 1,
      "username": "top_classifier",
      "totalPoints": 2500,
      "totalClassifications": 250,
      "accuracy": 89.5
    }
  ],
  "currentUserRank": {
    "rank": 15,
    "totalPoints": 1050
  },
  "pagination": {
    "page": 0,
    "size": 10,
    "totalPages": 5,
    "totalElements": 48
  }
}
```

Dashboard Endpoints

3. Admin Endpoints Summary

	Purpose	Endpoint
3.1	View Tasks	GET /dashboard/tasks
3.2	Create Task	POST /dashboard/tasks/create
3.3	Archive Task	POST /dashboard/tasks/archive/{taskId}
3.4	Update Task	PUT /dashboard/tasks/{taskId}
3.5	Activate Task	POST /dashboard/tasks/{taskId}/activate
3.6	Pause Task	POST /dashboard/tasks/{taskId}/pause
3.7	Task analytics	GET /dashboard/tasks/{taskId}/analytics
3.8	Export results	POST /dashboard/exports
3.9	View Gold Images	GET /dashboard/images/gold
4.0	Upload Gold Image	POST /dashboard/images/gold/upload
4.1	Delete Gold Image	DELETE /dashboard/images/gold/{imageId}
4.2	View Recipients List	GET /dashboard/recipients/
4.3	Create Recipients List	POST /dashboard/recipients/create
4.4	Delete Recipients List	DELETE /dashboard/recipients/{groupId}
4.5	Update Recipients List	PUT /dashboard/recipients/{groupId}/update
4.6	View Taxonomy	GET /dashboard/taxonomy

3.1 View Tasks

```
GET /api/v1/dashboard/tasks
Request:
Headers:
Authorization: Bearer <accessToken>
Body:
  (none)

Response:
{
  "tasks": [
    {
      "taskId": 7,
      "status": "ACTIVE",
      "name": "Asian Giant Hornet Identification",
      "description": "Identify Asian Giant Hornets in trap images",
      "targetSpecies": [{"name": "Vespa mandarinia",
                        "commonName": "Asian Giant Hornet",
                        "referenceImages": [
                          { "contentType": "image/jpeg",
                            "data": "<base64>",
                            "caption": "Dorsal view"
                          },
                          { "contentType": "image/jpeg",
                            "data": "<base64>",
                            "caption": "Side view"
                          }
                        ]
                      }],
      "experiments": [101, 102],
      "recipientGroups": [3, 7],
      "progress": {
        "totalImages": 1000,
        "imagesClassified": 320
      }
      "minClassificationsPerImage": 3,
      "consensusThreshold": 80.0
    }
  ]
}
```

3.2 Create Task

```
POST /api/v1/dashboard/tasks/create
```

Request:

Headers:

```
Authorization: Bearer <accessToken>
```

Body:

```
{  
    "name": "Asian Giant Hornet Identification",  
    "description": "Identify Asian Giant Hornets in trap images",  
    "targetSpecies": [{"name": "Vespa mandarinia",  
                      "commonName": "Asian Giant Hornet",  
                      "referenceImages": [  
                            { "contentType": "image/jpeg",  
                              "data": "<base64>",  
                              "caption": "Dorsal view"  
                            },  
                            { "contentType": "image/jpeg",  
                              "data": "<base64>",  
                              "caption": "Side view"  
                            }  
                          ]},  
    "experiments": [101, 102],  
    "recipientGroups": [3, 7],  
    "minClassificationsPerImage": 3,  
    "consensusThreshold": 80.0  
}
```

Response:

```
{  
  "tasks": [  
    {  
      "taskId": 7,  
      "status": "ACTIVE",  
      "name": "Asian Giant Hornet Identification",  
      "description": "Identify Asian Giant Hornets in trap images",  
      "targetSpecies": [{"name": "Vespa mandarinia",  
                        "commonName": "Asian Giant Hornet",  
                        "referenceImages": [  
                              { "contentType": "image/jpeg",  
                                "data": "<base64>",  
                                "caption": "Dorsal view"  
                              },  
                            ]}  
    }]
```

```
        {
            "contentType": "image/jpeg",
            "data": "<base64>",
            "caption": "Side view"
        }
    ],
    "experiments": [101, 102],
    "recipientGroups": [3, 7],
    "progress": {
        "totalImages": 1000 (sum of experiments images),
        "imagesClassified": 0
    }
}
]
```

3.3 Archive Task

```
POST /api/v1/dashboard/tasks/archive/{task-id}
```

Request:

Headers:

```
Authorization: Bearer <accessToken>
```

Response:

```
{
    "taskId": 7,
    "status": "ARCHIVED",
    "archivedAt": "2025-12-20T13:45:00Z"
}
```

3.4 Update Task

```
POST /api/v1/dashboard/tasks/archive/{task-id}
```

Request:

Headers:

```
Authorization: Bearer <accessToken>
```

Body:

```
{
    "status": "ACTIVE",
    "name": "Asian Giant Hornet Identification",
    "description": "Identify Asian Hornets in trap images",
```

```
"targetSpecies": [
  {
    "name": "Vespa mandarinia",
    "referenceImages": [
      {
        "contentType": "image/jpeg",
        "data": "<base64>",
        "caption": "Dorsal view"
      },
      {
        "contentType": "image/jpeg",
        "data": "<base64>",
        "caption": "Side view"
      }
    ]
  },
  {
    "experiments": [101, 102],
    "recipientGroups": [3, 7]
  }
]
```

Response:

```
{
  "taskId": 7,
  "status": "ACTIVE",
  "name": "Asian Giant Hornet Identification",
  "description": "Identify Asian Hornets in trap images",

  "targetSpecies": [
    {
      "name": "Vespa mandarinia",
      "referenceImages": [
        {
          "contentType": "image/jpeg",
          "data": "<base64>",
          "caption": "Dorsal view"
        }
      ]
    },
    {
      "experiments": [101, 102],
      "recipientGroups": [3, 7],
      "progress": {

```

```
        "totalImages": 1000,  
        "imagesClassified": 0  
    },  
  
    "updatedAt": "2025-12-20T14:55:00Z"  
}
```

3.5 Activate Task

```
POST /api/v1/dashboard/tasks/{taskId}/activate
```

Request:

Headers:

```
Authorization: Bearer <accessToken>
```

Body: (empty)

Response:

```
{  
    "taskId": 7,  
    "previousStatus": "DRAFT",  
    "currentStatus": "ACTIVE",  
    "activatedAt": "2025-12-20T15:10:00Z"  
}
```

🚦 Allowed State Transitions

From	To	Allowed
DRAFT	ACTIVE	✓
PAUSED	ACTIVE	✓
ACTIVE	ACTIVE	✗
COMPLETED	ACTIVE	✗
ARCHIVED	ACTIVE	✗

3.6 Pause Task

```
POST /api/v1/dashboard/tasks/{taskId}/pause
```

Request:

Headers:

```
Authorization: Bearer <accessToken>
```

Response:

```
{  
  "taskId": 7,  
  "previousStatus": "ACTIVE",  
  "currentStatus": "PAUSED",  
  "pausedAt": "2025-12-20T15:25:00Z"  
}
```

3.7 Task Analytics

```
GET /dashboard/tasks/{taskId}/analytics
```

Request:

Headers:

```
Authorization: Bearer <accessToken>
```

Query Params (optional):

```
?includePerSpecies=true  
&includeTimeSeries=true
```

Response:

```
{  
  "taskId": 7,  
  "status": "ACTIVE",  
  
  "progress": {  
    "totalImages": 1000,  
    "imagesClassified": 450,  
    "completedImages": 320,  
    "percentComplete": 32.0  
  },  
  
  "consensus": {  
    "overallAverage": 84.5,  
    "lowConsensusImages": 15,  
    "threshold": 80.0  
  },  
}
```

```
"speciesAnalytics": [
  {
    "name": "Vespa mandarinia",
    "classificationCount": 720,
    "agreementRate": 88.1,
    "confusionMatrix": {
      "truePositive": 410,
      "falsePositive": 38,
      "falseNegative": 52,
      "trueNegative": 220
    }
  },
  {
    "name": "Vespa orientalis",
    "classificationCount": 630,
    "agreementRate": 81.7
  }
],
"participation": {
  "activeUsers": 45,
  "totalClassifications": 1350,
  "averageClassificationsPerUser": 30,
  "medianResponseTimeMs": 2800
},
"quality": {
  "averageCredibility": 72.3,
  "expertAgreement": 86.4,
  "lowQualityUsers": 3
},
"timeSeries": [
  {
    "date": "2025-12-15",
    "classifications": 120,
    "consensusReached": 35
  }
],
"generatedAt": "2025-12-20T15:45:00Z"
}
```

🕒 Availability by Task Status

Status Analytics

DRAFT	
ACTIVE	
PAUSED	
COMPLETED	
ARCHIVED	(read-only)

Optional Extensions:

- **Per-Experiment Breakdown**

?groupBy=experiment

- **Per-Recipient Group**

?groupBy=recipientGroup

- **Export-ready Analytics**

?format=EXPORT

Dashboard Analytics Endpoints

Endpoint	Purpose
GET /dashboard/tasks/{taskId}/analytics	Task analytics
GET /dashboard/tasks/{taskId}/analytics/species	Per-species
GET /dashboard/tasks/{taskId}/analytics/users	User performance
GET /dashboard/tasks/{taskId}/analytics/time	Time series

3.8 Export Results

Export Endpoint Summary

Purpose	Endpoint
Create export	POST /dashboard/exports
Check status	GET /dashboard/exports/{exportId}
Download export	via downloadUrl
(Optional) Proxy download	GET /dashboard/exports/{exportId}/download
Cancel export	DELETE /dashboard/exports/{exportId}

Purpose

Creates an **asynchronous export job** for task data.

- **Admin-only**
- Long-running
- Non-blocking
- Produces downloadable scientific datasets

```
POST /dashboard/exports
Request:
Headers:
Authorization: Bearer <accessToken>
Body:
{
  "taskId": 7,
  "format": "CSV",
  "scope": "CLASSIFICATIONS",

  "filters": {
    "minConsensus": 80.0,
    "minCredibility": 60.0,
    "species": ["Vespa mandarinia"],
    "dateRange": {
      "from": "2025-12-01",
      "to": "2025-12-20"
    }
}
```

```
},
  "include": {
    "speciesMetadata": true,
    "referenceImages": false,
    "expertLabels": true,
    "userMetadata": false
  }
}

Response:
{
  "exportId": "exp_123456",
  "status": "QUEUED",
  "createdAt": "2025-12-20T16:05:00Z",
  "estimatedCompletion": "2025-12-20T16:15:00Z"
}
HTTP status: 202 Accepted
```

3.9 View Gold Images

```
GET /api/v1/dashboard/images/gold
Request:
Headers:
Authorization: Bearer <accessToken>
Query Parameters (optional):
?taskId=7
?experimentId=101
?page=1
&pageSize=50
```

```
Response:
{
  "page": 1,
  "pageSize": 50,
  "totalPages": 2,
  "totalImages": 75,
  "images": [
    {
      "imageId": 501,
      "taskId": 7,
      "experimentId": 101,
      "species": "Vespa mandarinia",
```

```
        "imageUrl": "https://swipelab.com/gold/501.jpg",
        "thumbnailUrl": "https://swipelab.com/gold/thumbs/501.jpg",
        "caption": "Asian Giant Hornet - dorsal view",
        "uploadedAt": "2025-12-15T14:30:00Z"
    }
]
}
```

4.0 Add Gold Image

```
POST /api/v1/dashboard/images/gold/upload
Request:
Headers:
Authorization: Bearer <accessToken>
Content-Type: multipart/form-data
Form Data:
  file: <image file>
  taskId: 7
  experimentId: 101
  species: Vespa mandarinia
  caption: "Dorsal view"
  isPositiveExample: true

Response:
{
  "imageId": 502,
  "taskId": 7,
  "experimentId": 101,
  "species": "Vespa mandarinia",
  "imageUrl": "https://swipelab.com/gold/502.jpg",
  "thumbnailUrl": "https://swipelab.com/gold/thumbs/502.jpg",
  "caption": "Dorsal view",
  "isPositiveExample": true,
  "uploadedAt": "2025-12-20T16:45:00Z"
}
```

4.1 Delete Gold Image

```
DELETE /api/v1/dashboard/images/gold/{imageId}
Request:
Headers:
Authorization: Bearer <accessToken>
```

```
Response:  
{  
  "imageId": 502,  
  "status": "DELETED",  
  "deletedAt": "2025-12-20T17:00:00Z"  
}
```

4.2 View Recipients List

```
GET /api/v1/dashboard/recipients  
Request:  
Headers:  
Authorization: Bearer <accessToken>  
Query Parameters (optional):  
?page=1  
&pageSize=50
```

```
Response:  
{  
  "page": 1,  
  "pageSize": 50,  
  "totalPages": 1,  
  "totalGroups": 3,  
  "recipientGroups": [  
    {  
      "groupId": 1,  
      "name": "Volunteer Group A",  
      "userCount": 45,  
      "usernames": ["john_doe", "alice_smith", "..."],  
      "createdAt": "2025-12-15T14:30:00Z"  
    }  
  ]  
}
```

4.3 Create Recipients List

```
POST /api/v1/dashboard/recipients/create  
Request:  
Headers:  
Authorization: Bearer <accessToken>  
Body:
```

```
{  
  "name": "New Volunteer Group",  
  "usernames": ["john_doe", "alice_smith", "bob_jones"]  
}
```

Response:

```
{  
  "groupId": 3,  
  "name": "New Volunteer Group",  
  "userCount": 3,  
  "usernames": ["john_doe", "alice_smith", "bob_jones"],  
  "createdAt": "2025-12-20T17:15:00Z"  
}
```

4.4 Delete Recipients List

DELETE /api/v1/dashboard/recipients/{groupId}

Request:

Headers:

Authorization: Bearer <accessToken>

Response:

```
{  
  "taskId": 7,  
  "previousStatus": "ACTIVE",  
  "currentStatus": "PAUSED",  
  "pausedAt": "2025-12-20T15:25:00Z"  
}
```

Commented [vv6]: 💬 Backend Notes

1.Validation

- Each username must exist
- No duplicates in a group

2.Soft delete

Recommended for audit

3.Permissions

Only admins can create/view/delete groups

4.5 Update Recipients List

PUT /api/v1/dashboard/recipients/{groupId}/update

Request:

Headers:

Authorization: Bearer <accessToken>

Content-Type: application/json

Body:

```
{  
  "addUsernames": ["new_user1", "new_user2"],  
  "removeUsernames": ["old_user1"]  
}
```

Commented [vv7]: 💬 Backend Notes

1.Validation

(1)

- Added usernames must exist and not already be in the group
- Removed usernames must exist in the group
- Cannot remove all users (optional, depending on business rules)

1.Audit

(1)

(1)Log which users were added/removed

1.Permissions

(1)

(1)Only admins can update recipient groups

1.Soft delete

(1)

(1)Optional: if a user was deleted from the system, automatically remove from all groups

```
Response:  
{  
    "groupId": 3,  
    "name": "New Volunteer Group",  
    "userCount": 4,  
    "usernames": ["john_doe", "alice_smith", "new_user1", "new_user2"],  
    "updatedAt": "2025-12-20T18:00:00Z"  
}
```

4.6 View Taxonomy

```
GET /api/v1/dashboard/taxonomy  
Request:  
Headers:  
Authorization: Bearer <accessToken>  
Query Parameters (optional):  
?level=species // class | order | family | genus | species  
?parent=<parent_id> // filter by parent id
```

```
Response:  
{  
    "class": "Insecta",  
    "orders": [  
        {  
            "order": "Hymenoptera",  
            "families": [  
                {  
                    "family": "Vespidae",  
                    "genera": [  
                        {  
                            "genus": "Vespa",  
                            "species": [  
                                { "species_id": 1234, "name": "Vespa mandarinia" },  
                                { "species_id": 1235, "name": "Vespa orientalis" }  
                            ]  
                        ]  
                    ]  
                ]  
            }  
        ]  
    }  
}
```

5. User

	Purpose	Endpoint
5.1	My tasks	GET /dashboard/my-tasks
5.2	Task details	GET /dashboard/my-tasks/{id}
5.3	Play Task	GET /dashboard/my-tasks/{id}/play

5.1 My Tasks

```
GET /api/v1/dashboard/my-tasks
Request:
Headers:
Authorization: Bearer <accessToken>
Query Parameters (optional):
?page=1
&pageSize=20
Body:
(none)

Response:
{
  "page": 1,
  "pageSize": 20,
  "totalPages": 2,
  "totalTasks": 25,
  "tasks": [
    {
      "taskId": 7,
      "name": "Asian Giant Hornet Identification",
      "description": "Identify Asian Giant Hornets in trap images",
      "species": [
        {
          "name": "Vespa mandarinia",
          "referenceImages": [
            {
              "imageUrl": "https://swipelab.com/ref1.jpg",
              "caption": "Dorsal view"
            }
          ]
        }
      ]
    }
  ]
}
```

```
        },
        {
            "imageUrl": "https://swipelab.com/ref2.jpg",
            "caption": "Side view"
        }
    ]
},
],
"totalImages": 100,
"imagesClassified": 32
}
]
```

5.2 Task Details

GET /api/v1/dashboard/my-tasks/{taskId}

Request:

Headers:

Authorization: Bearer <accessToken>

Body:

(none)

Response:

```
{
    "taskId": 7,
    "name": "Asian Giant Hornet Identification",
    "description": "Identify Asian Giant Hornets in trap images",
    "species": [
        {
            "name": "Vespa mandarinia",
            "referenceImages": [
                {
                    "imageUrl": "https://swipelab.com/ref1.jpg",
                    "caption": "Dorsal view"
                }
            ]
        }
    ],
    "totalImages": 100,
    "imagesClassified": 32,
    "assignedToUser": true,
    "status": "ACTIVE",
```

```
        "createdAt": "2025-12-15T14:30:00Z",
        "deadline": "2026-01-10T23:59:59Z"
    }
```

5.3 Play Task

```
GET /api/v1/dashboard/my-tasks/{taskId}/play
```

Request:

Headers:

```
Authorization: Bearer <accessToken>
```

Query Parameters (optional):

```
?count=10 // number of images to retrieve
```

Body:

(none)

Response:

```
{
    "taskId": 7,
    "species": [
        {
            "scientificName": "Vespa mandarinia",
            "commonName": "Asian Giant Hornet",
            "referenceImages": [
                {
                    "imageUrl": "https://stardbi.com/ref1.jpg",
                    "caption": "Dorsal view"
                }
            ]
        }
    ],
    "images": [
        {
            "imageId": 1234,
            "imageBuffer": "<base64-encoded-image-data>",
            "contentType": "image/jpeg",
            "question": "Is this a Vespa mandarinia?",
            "taskId": 7,
            "species": "Vespa mandarinia"
        },
        {
            "imageId": 1235,
            "imageBuffer": "<base64-encoded-image-data>",
            "contentType": "image/jpeg",
            "question": "Is this a Vespa mandarinia?"
        }
    ]
}
```

Commented [pvw8]: This method should set the User's batch settings to that specific task (he will only get images relevant to this task)

```
        "question": "Is this a Vespa mandarinia?",  
        "taskId": 7,  
        "species": "Vespa mandarinia"  
    }  
]
```