

Pay Rate Configuration API Documentation

For Front-End Developers

Overview

This API manages pay rate configurations for farm staff. It provides endpoints to view default Horticulture Award 2025 rates and update pay rates for individual staff or groups. All update operations require Admin authentication.

Base URL

```
/api/payrates
```

Authentication

- **GET endpoints:** No authentication required
- **PUT/POST endpoints:** Require Admin role with JWT token in Authorization header: `Bearer <token>`

API Endpoints Summary

Method	Endpoint	Auth	Description
GET	<code>/api/payrates/defaults</code>	No	Get all default Award rates
PUT	<code>/api/payrates/staff/{id}/payrates</code>	Admin	Update pay rate for one staff
PUT	<code>/api/payrates/bulk/{role}/{contractType}</code>	Admin	Bulk update pay rates by role/contract
POST	<code>/api/payrates/initialize-defaults</code>	Admin	Initialize all staff with default rates

1. Get All Default Pay Rates

```
GET /api/payrates/defaults
```

Description: Retrieves all default Horticulture Award 2025 pay rates for different roles and contract types.

Auth Required: No

Response:

```
{
  "Worker": {
    "Full-time": {
      "standardRate": 25.00,
      "overtimeRate": 37.50,
      "weekendRate": 50.00
    },
    "Part-time": {
      "standardRate": 25.00,
      "overtimeRate": 37.50,
      "weekendRate": 50.00
    },
    "Casual": {
      "standardRate": 31.25,
      "overtimeRate": 46.87,
      "weekendRate": 62.50
    }
  },
  "Manager": {
    "Full-time": {
      "standardRate": 35.00,
      "overtimeRate": 52.50,
      "weekendRate": 70.00
    },
    "Part-time": {
      "standardRate": 35.00,
      "overtimeRate": 52.50,
      "weekendRate": 70.00
    },
    "Casual": {
      "standardRate": 43.75,
      "overtimeRate": 65.62,
      "weekendRate": 87.50
    }
  }
}
```

Example:

```
curl http://localhost:5000/api/payrates/defaults
```

2. Update Individual Staff Pay Rates

```
PUT /api/payrates/staff/{id}/payrates
Authorization: Bearer <jwt-token>
Content-Type: application/json
```

Description: Updates pay rates for a specific staff member. Admin access required.

Auth Required: Yes (Admin role)

URL Parameters:

- `id` (integer): Staff ID

Request Body:

```
{
  "standardPayRate": 26.00,
  "overtimePayRate": 39.00
}
```

Response (Success):

```
{
  "message": "Staff pay rates updated successfully",
  "staff": "{...updated staff object...}"
}
```

Response (Error - Not Admin):

```
{
  "message": "Only admin users can update pay rates"
}
```

Example:

```
curl -X PUT http://localhost:5000/api/payrates/staff/5/payrates \
-H "Authorization: Bearer eyJhbGc..." \
-H "Content-Type: application/json" \
-d '{
  "standardPayRate": 28.00,
  "overtimePayRate": 42.00
}'
```

Use Cases:

- Increase pay for individual staff member
- Set custom pay rate for experienced worker
- Adjust rates for performance bonuses

3. Bulk Update Pay Rates by Role and Contract Type

```
PUT /api/payrates/bulk/{role}/{contractType}
Authorization: Bearer <jwt-token>
Content-Type: application/json
```

Description: Updates pay rates for all staff with a specific role and contract type combination. Admin access required.

Auth Required: Yes (Admin role)

URL Parameters:

- **role** (string): "Worker" or "Manager"
- **contractType** (string): "Full-time", "Part-time", or "Casual"

Request Body:

```
{
  "standardPayRate": 27.00,
  "overtimePayRate": 40.50
}
```

Response (Success):

```
{
  "message": "Successfully updated 10 staff members with role 'Worker' and
contract type 'Full-time'",
  "staff": "[...array of updated staff...]"
}
```

Example:

```
curl -X PUT http://localhost:5000/api/payrates/bulk/Worker/Full-time \
-H "Authorization: Bearer eyJhbGc..." \
-H "Content-Type: application/json" \
-d '{
  "standardPayRate": 27.00,
  "overtimePayRate": 40.50
}'
```

Use Cases:

- Apply new Award rates to all workers in a category
- Increase rates for all casual staff

- Update all manager pay rates

Valid Combinations:

Role	Contract Type
Worker	Full-time
Worker	Part-time
Worker	Casual
Manager	Full-time
Manager	Part-time
Manager	Casual

4. Initialize Default Pay Rates

```
POST /api/payrates/initialize-defaults
Authorization: Bearer <jwt-token>
```

Description: Initializes default Horticulture Award 2025 pay rates for ALL staff members based on their role and contract type. This operation updates every staff member in the system. Admin access required.

Auth Required: Yes (Admin role)

Request Body: None

Response (Success):

```
{
  "message": "Successfully updated 25 staff members with default Horticulture Award 2025 pay rates",
  "staff": "[...array of all updated staff...]"
}
```

Example:

```
curl -X POST http://localhost:5000/api/payrates/initialize-defaults \
-H "Authorization: Bearer eyJhbGc..."
```

Use Cases:

- Initial system setup
- Apply new Award rates to entire workforce

- Reset all pay rates to standard values
- ⚠ Warning:** This operation affects ALL staff members. Use with caution!

Default Pay Rate Reference

Horticulture Award 2025 Rates

Worker Rates

Contract Type	Standard Rate	Overtime Rate	Weekend Rate
Full-time	\$25.00/hr	\$37.50/hr	\$50.00/hr
Part-time	\$25.00/hr	\$37.50/hr	\$50.00/hr
Casual	\$31.25/hr	\$46.87/hr	\$62.50/hr

Manager Rates

Contract Type	Standard Rate	Overtime Rate	Weekend Rate
Full-time	\$35.00/hr	\$52.50/hr	\$70.00/hr
Part-time	\$35.00/hr	\$52.50/hr	\$70.00/hr
Casual	\$43.75/hr	\$65.62/hr	\$87.50/hr

Notes:

- Overtime Rate = Standard Rate × 1.5
- Weekend Rate = Standard Rate × 2.0
- Casual rates include 25% casual loading

Error Responses

401 Unauthorized

```
{
  "message": "Authentication required"
}
```

Cause: No JWT token provided or token is invalid

403 Forbidden

```
{
  "message": "Only admin users can update pay rates"
}
```

```
}
```

Cause: User is authenticated but not an Admin

400 Bad Request

```
{  
  "message": "Failed to update staff pay rates"  
}
```

Cause: Invalid request data or staff not found

Authentication Guide

Getting JWT Token

First, login as an admin user:

```
POST /api/staffs/login  
{  
  "email": "admin@farm.com",  
  "password": "your-password"  
}
```

Response will include JWT token. Use it in subsequent requests:

```
Authorization: Bearer <your-jwt-token>
```

Testing Examples

Testing with Postman

1. Get Default Rates

- Method: **GET**
- URL: **http://localhost:5000/api/payrates/defaults**
- Auth: None
- Send Request

2. Update Individual Staff

- Method: **PUT**

- URL: `http://localhost:5000/api/payrates/staff/1/payrates`
- Auth: Bearer Token (from login)
- Body (raw JSON):

```
{
  "standardPayRate": 26.00,
  "overtimePayRate": 39.00
}
```

3. Bulk Update Workers

- Method: `PUT`
- URL: `http://localhost:5000/api/payrates/bulk/Worker/Full-time`
- Auth: Bearer Token
- Body (raw JSON):

```
{
  "standardPayRate": 27.00,
  "overtimePayRate": 40.50
}
```

4. Initialize Defaults

- Method: `POST`
- URL: `http://localhost:5000/api/payrates/initialize-defaults`
- Auth: Bearer Token
- Body: None

JavaScript Integration Examples

Using Axios

```
import axios from 'axios';

// Get JWT token (assume you have it from login)
const token = localStorage.getItem('authToken');

// 1. Get all default rates (no auth needed)
const getDefaults = async () => {
  const response = await axios.get('/api/payrates/defaults');
  console.log(response.data);
};

// 2. Update individual staff pay rate (admin only)
const updateStaffRate = async (staffId, standardRate, overtimeRate) => {
```



```
try {
  const response = await axios.put(
    `/api/payrates/staff/${staffId}/payrates`,
    {
      standardPayRate: standardRate,
      overtimePayRate: overtimeRate
    },
    {
      headers: {
        'Authorization': `Bearer ${token}`,
        'Content-Type': 'application/json'
      }
    }
  );
  console.log('Success:', response.data.message);
} catch (error) {
  console.error('Error:', error.response?.data?.message);
}
};

// 3. Bulk update by role and contract (admin only)
const bulkUpdateRates = async (role, contractType, standardRate, overtimeRate) => {
  try {
    const response = await axios.put(
      `/api/payrates/bulk/${role}/${contractType}`,
      {
        standardPayRate: standardRate,
        overtimePayRate: overtimeRate
      },
      {
        headers: {
          'Authorization': `Bearer ${token}`,
          'Content-Type': 'application/json'
        }
      }
    );
    console.log(response.data.message);
  } catch (error) {
    console.error('Error:', error.response?.data?.message);
  }
};

// 4. Initialize defaults for all staff (admin only)
const initializeDefaults = async () => {
  try {
    const response = await axios.post(
      '/api/payrates/initialize-defaults',
      {},
      {
        headers: {
          'Authorization': `Bearer ${token}`
        }
      }
    );
  }
}
```

```

    );
    console.log(response.data.message);
  } catch (error) {
    console.error('Error:', error.response?.data?.message);
  }
};

```

Using Fetch API

```

// Get default rates
const getDefaults = async () => {
  const response = await fetch('/api/payrates/defaults');
  const data = await response.json();
  console.log(data);
};

// Update staff pay rate (with auth)
const updateStaffRate = async (staffId, standardRate, overtimeRate) => {
  const token = localStorage.getItem('authToken');

  const response = await fetch(`/api/payrates/staff/${staffId}/payrates`, {
    method: 'PUT',
    headers: {
      'Authorization': `Bearer ${token}`,
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({
      standardPayRate: standardRate,
      overtimePayRate: overtimeRate
    })
  });

  if (response.ok) {
    const data = await response.json();
    console.log(data.message);
  } else {
    const error = await response.json();
    console.error(error.message);
  }
};

```

React Component Example

```

import { useState } from 'react';
import axios from 'axios';

function PayRateManager() {
  const [defaultRates, setDefaultRates] = useState(null);

```

```
const [loading, setLoading] = useState(false);
const [message, setMessage] = useState('');

// Get default rates
const loadDefaults = async () => {
  const response = await axios.get('/api/payrates/defaults');
  setDefaultRates(response.data);
};

// Update individual staff
const updateStaff = async (staffId, standardRate, overtimeRate) => {
  setLoading(true);
  try {
    const token = localStorage.getItem('authToken');
    const response = await axios.put(
      `/api/payrates/staff/${staffId}/payrates`,
      { standardPayRate: standardRate, overtimePayRate: overtimeRate },
      { headers: { 'Authorization': `Bearer ${token}` } }
    );
    setMessage(response.data.message);
  } catch (error) {
    setMessage(error.response?.data?.message || 'Error updating rates');
  } finally {
    setLoading(false);
  }
};

// Bulk update
const bulkUpdate = async (role, contractType, standardRate, overtimeRate) => {
  setLoading(true);
  try {
    const token = localStorage.getItem('authToken');
    const response = await axios.put(
      `/api/payrates/bulk/${role}/${contractType}`,
      { standardPayRate: standardRate, overtimePayRate: overtimeRate },
      { headers: { 'Authorization': `Bearer ${token}` } }
    );
    setMessage(response.data.message);
  } catch (error) {
    setMessage(error.response?.data?.message || 'Error updating rates');
  } finally {
    setLoading(false);
  }
};

// Initialize defaults
const initializeDefaults = async () => {
  if (!window.confirm('This will update ALL staff members. Continue?')) return;

  setLoading(true);
  try {
    const token = localStorage.getItem('authToken');
    const response = await axios.post(
      '/api/payrates/initialize-defaults',
```

```

    {}},
    { headers: { 'Authorization': `Bearer ${token}` }}
  );
  setMessage(response.data.message);
} catch (error) {
  setMessage(error.response?.data?.message || 'Error initializing rates');
} finally {
  setLoading(false);
}
};

return (
  <div>
    <h2>Pay Rate Manager</h2>

    <button onClick={loadDefaults}>Load Default Rates</button>
    <button onClick={initializeDefaults} disabled={loading}>
      Initialize All Defaults
    </button>

    {message && <p>{message}</p>}
    {loading && <p>Processing...</p>}

    {defaultRates && (
      <pre>{JSON.stringify(defaultRates, null, 2)}</pre>
    )}
  </div>
);
}

```

Workflow Examples

Scenario 1: New System Setup

1. Login as admin
2. Call `POST /api/payrates/initialize-defaults`
3. All staff get default Award rates based on their Role/ContractType

Scenario 2: Annual Pay Increase

1. Login as admin
2. For each role/contract combination:
 - Call `PUT /api/payrates/bulk/{role}/{contractType}` with new rates
3. Verify updates with `GET /api/payrates/defaults` (to see what defaults are)

Scenario 3: Individual Adjustment

1. Login as admin
2. Identify staff ID needing adjustment
3. Call `PUT /api/payrates/staff/{id}/payrates` with custom rates

Troubleshooting

Common Issues

Issue	Cause	Solution
401 Unauthorized	No token or expired	Login again to get fresh token
403 Forbidden	Not an admin	Login with admin account
400 Bad Request	Invalid staff ID	Verify staff exists in database
400 Bad Request	Invalid role/contract	Use valid combinations (see table above)

Validation Checklist

- ☒ User is authenticated (has JWT token)
- ☒ User has Admin role
- ☒ Staff ID exists (for individual updates)
- ☒ Role is "Worker" or "Manager"
- ☒ ContractType is "Full-time", "Part-time", or "Casual"
- ☒ Pay rates are positive numbers

Related APIs

- **PayRoll API:** [/api/payroll](#) - Calculate staff payroll (uses pay rates from this API)
- **Staff API:** [/api/staffs](#) - Manage staff information
- **Login API:** [/api/staffs/login](#) - Get authentication token

Support

For issues or questions:

- Check this documentation
- Verify authentication and permissions
- Contact the backend development team

Last Updated: January 2025
API Version: 1.0
Compliance: Horticulture Award 2025