

PayRoll Calculation API Documentation

For Front-End Developers

Overview

This API calculates weekly payroll for farm staff based on clock-in/out events. It implements Horticulture Award 2025 compliant calculations including:

- Ordinary hours (38 hours/week standard)
- Daily overtime (after 8 hours/day @ 1.5x)
- Weekly overtime (first 2 hours @ 1.5x, additional @ 2.0x)
- Weekend rates (Saturday/Sunday @ 2.0x)
- PAYG tax withholding based on ATO 2024-25 tax tables
- Employer superannuation (11.5%)

Base URL

```
/api/payroll
```

Authentication

Currently **No authentication required** for payroll calculations.

API Endpoints

Calculate Weekly Payroll

```
GET /api/payroll/calculate?staffId={staffId}&mondayDate={mondayDate}&isSpecialPayRate={true/false}
```

Description: Calculates complete weekly payroll for a staff member based on their clock-in/out events for a specific week.

Auth Required: No

Request Format

Query Parameters (URL)

```
/api/payroll/calculate?staffId=1&mondayDate=2024-12-30&isSpecialPayRate=true
```

Request Parameters

Parameter	Type	Required	Default	Description
staffId	integer	Yes	-	The ID of the staff member
mondayDate	string	Yes	-	The Monday date of the week (format: yyyy-MM-dd) MUST be a Monday
isSpecialPayRate	boolean	No	false	true: Use default Horticulture Award rates false: Use staff's current pay rate

Response Format

Success Response (200 OK)

```
{
  "staffId": 1,
  "staffName": "John Doe",
  "weekStartDate": "2024-12-30T00:00:00",
  "totalHoursWorked": 43.5,
  "grossWeeklyPay": 1400.00,
  "annualIncome": 72800.00,
  "annualTax": 12628.00,
  "weeklyPAYG": 243.00,
  "netPay": 1157.00,
  "employerSuperannuation": 161.00
}
```

Response Fields

Field	Type	Description
staffId	integer	Staff member ID
staffName	string	Full name of staff member
weekStartDate	datetime	Monday date of the payroll week
totalHoursWorked	decimal	Total hours worked (after breaks & rounding)
grossWeeklyPay	decimal	Total weekly pay before tax
annualIncome	decimal	Projected annual income (weekly × 52)
annualTax	decimal	Projected annual tax
weeklyPAYG	decimal	Weekly PAYG tax withholding
netPay	decimal	Take-home pay (Gross - PAYG)

Field	Type	Description
employerSuperannuation	decimal	Employer super contribution (11.5%)

Error Responses

400 Bad Request - Invalid Staff ID

```
{
  "message": "Staff ID must be greater than 0"
}
```

400 Bad Request - Missing Monday Date

```
{
  "message": "Monday date is required"
}
```

400 Bad Request - Invalid Date (Not Monday)

```
{
  "message": "Error: The provided date is not a Monday",
  "weekStartDate": "2024-12-31T00:00:00"
}
```

400 Bad Request - Invalid Date Format

```
{
  "message": "Invalid date format. Use format: yyyy-MM-dd"
}
```

404 Not Found - Staff Not Found

```
{
  "message": "Staff with ID 9999 not found"
}
```

Usage Examples

Example 1: Calculate Payroll with Current Pay Rate

Request:

```
GET /api/payroll/calculate?staffId=1&mondayDate=2024-12-30
```

Response:

```
{
  "staffId": 1,
  "staffName": "John Smith",
  "weekStartDate": "2024-12-30T00:00:00",
  "totalHoursWorked": 40.0,
  "grossWeeklyPay": 1000.00,
  "annualIncome": 52000.00,
  "annualTax": 8139.00,
  "weeklyPAYG": 156.52,
  "netPay": 843.48,
  "employerSuperannuation": 115.00
}
```

Example 2: Calculate Payroll with Default Award Rates

Request:

```
GET /api/payroll/calculate?staffId=1&mondayDate=2024-12-30&isSpecialPayRate=true
```

Response: Uses default Horticulture Award 2025 rates based on staff's Role and ContractType:

- Worker Full-time/Part-time: \$25.00/hr
- Worker Casual: \$31.25/hr
- Manager Full-time/Part-time: \$35.00/hr
- Manager Casual: \$43.75/hr

Example 3: Explicitly Disable Special Pay Rate

Request:

```
GET /api/payroll/calculate?staffId=1&mondayDate=2024-12-30&isSpecialPayRate=false
```

Example 4: Error - Invalid Date (Not Monday)

Request:

```
GET /api/payroll/calculate?staffId=1&mondayDate=2024-12-31
```

Response (400 Bad Request):

```
{
  "message": "Error: The provided date is not a Monday",
  "weekStartDate": "2024-12-31T00:00:00"
}
```

Payroll Calculation Logic

Step 1: Calculate Hours Worked

- Query all clock-in/out events for the week (Monday to Sunday)
- Pair clock-in with clock-out events
- **Round to nearest 5 minutes** (Award compliance)
- **Deduct 30 minutes unpaid break** if shift > 5 hours

Step 2: Calculate Gross Pay

```
Gross Pay = Ordinary Hours × Base Rate
            + Daily Overtime Hours × 1.5 × Base Rate
            + Weekly Overtime (first 2 hrs) × 1.5 × Base Rate
            + Weekly Overtime (additional) × 2.0 × Base Rate
            + Weekend/Holiday Hours × 2.0 × Base Rate
```

Overtime Rules:

- **Daily OT:** Hours beyond 8 per day @ 1.5x
- **Weekly OT:** Hours beyond 38 per week
 - First 2 hours @ 1.5x
 - Additional hours @ 2.0x
- **Weekend:** Saturday/Sunday @ 2.0x

Step 3: Calculate Annual Income

```
Annual Income = Gross Weekly Pay × 52
```

Step 4: Calculate PAYG Tax (ATO 2024-25 Tax Tables)

Annual Income Range	Tax Calculation
---------------------	-----------------

Annual Income Range	Tax Calculation
\$0 – \$18,200	Tax = \$0
\$18,201 – \$45,000	Tax = 16% × (Income – \$18,200)
\$45,001 – \$135,000	Tax = \$4,288 + 30% × (Income – \$45,000)
\$135,001 – \$190,000	Tax = \$31,288 + 37% × (Income – \$135,000)
\$190,001+	Tax = \$51,738 + 45% × (Income – \$190,000)

$$\text{Weekly PAYG} = \text{Annual Tax} \div 52$$

Step 5: Calculate Net Pay

$$\text{Net Pay} = \text{Gross Pay} - \text{PAYG Withholding}$$

Step 6: Calculate Employer Superannuation

$$\text{Superannuation} = \text{Gross Pay} \times 11.5\%$$

Payroll Calculation Example

Scenario

- Staff: Full-time Worker
- Base Rate: \$25/hr
- Week: Monday Dec 30, 2024 - Sunday Jan 5, 2025

Hours Worked:

Day	Hours	Type	Calculation
Mon	10 hrs	8 ord + 2 daily OT	$8 \times \$25 + 2 \times \$37.50 = \$275$
Tue	8 hrs	Ordinary	$8 \times \$25 = \200
Wed	8 hrs	Ordinary	$8 \times \$25 = \200
Thu	8 hrs	Ordinary	$8 \times \$25 = \200
Fri	8 hrs	Ordinary	$8 \times \$25 = \200
Sun	4 hrs	Weekend @ 2x	$4 \times \$50 = \200
Total	46 hrs	(43.5 after breaks)	

Payroll Breakdown:

1. **Ordinary Hours (38 hrs):** $38 \times \$25 = \950.00
2. **Monday Daily OT (2 hrs):** $2 \times \$37.50 = \75.00
3. **Weekly OT (4 hrs beyond 38):**

◦ First 2 hrs @ 1.5x: $2 \times \$37.50 = \75.00

◦ Next 2 hrs @ 2.0x: $2 \times \$50.00 = \100.00
4. **Sunday (4 hrs @ 2.0x):** $4 \times \$50.00 = \200.00

Total Gross Pay: \$1,400.00

Tax Calculation:

- Annual Income: $\$1,400 \times 52 = \$72,800$
- Tax Bracket: \$45,001 – \$135,000
- Tax: $\$4,288 + 30\% \times (\$72,800 - \$45,000) = \$12,628$
- Weekly PAYG: $\$12,628 \div 52 = \$243/\text{week}$

Net Pay: $\$1,400 - \$243 = \$1,157$

Superannuation: $\$1,400 \times 11.5\% = \161

Monday Dates Reference

Here are upcoming Monday dates for testing:

Date	Day of Week
2024-12-30	Monday <input checked="" type="checkbox"/>
2025-01-06	Monday <input checked="" type="checkbox"/>
2025-01-13	Monday <input checked="" type="checkbox"/>
2025-01-20	Monday <input checked="" type="checkbox"/>
2025-01-27	Monday <input checked="" type="checkbox"/>
2024-12-31	Tuesday <input checked="" type="checkbox"/>

Testing with cURL

Basic Payroll Calculation

```
curl "http://localhost:5000/api/payroll/calculate?staffId=1&mondayDate=2024-12-30"
```

With Special Pay Rate

```
curl "http://localhost:5000/api/payroll/calculate?staffId=1&mondayDate=2024-12-30&isSpecialPayRate=true"
```

Explicitly Disable Special Rate

```
curl "http://localhost:5000/api/payroll/calculate?staffId=1&mondayDate=2024-12-30&isSpecialPayRate=false"
```

Testing with Browser

Simply paste the URL into your browser's address bar:

```
http://localhost:5000/api/payroll/calculate?staffId=1&mondayDate=2024-12-30&isSpecialPayRate=true
```

The JSON response will be displayed directly in the browser.

Testing with JavaScript/Axios

```
// Basic payroll calculation
const response = await axios.get('/api/payroll/calculate', {
  params: {
    staffId: 1,
    mondayDate: '2024-12-30'
  }
});

console.log('Gross Pay:', response.data.grossWeeklyPay);
console.log('Net Pay:', response.data.netPay);
console.log('Super:', response.data.employerSuperannuation);

// With special pay rate
const responseSpecial = await axios.get('/api/payroll/calculate', {
  params: {
    staffId: 1,
    mondayDate: '2024-12-30',
    isSpecialPayRate: true
  }
});
```

Using Fetch API


```
// Basic request
const response = await fetch(
  '/api/payroll/calculate?staffId=1&mondayDate=2024-12-30'
);
const payroll = await response.json();

// With special pay rate
const url = new URL('/api/payroll/calculate', window.location.origin);
url.searchParams.append('staffId', 1);
url.searchParams.append('mondayDate', '2024-12-30');
url.searchParams.append('isSpecialPayRate', true);

const response2 = await fetch(url);
const payroll2 = await response2.json();
```

Testing with Postman

1. Create New Request

- Method: `GET`
- URL: `http://localhost:5000/api/payroll/calculate`

2. Add Query Parameters (Params tab)

Key	Value
<code>staffId</code>	<code>1</code>
<code>mondayDate</code>	<code>2024-12-30</code>
<code>isSpecialPayRate</code>	<code>true</code>

3. Send Request

Important Notes

Date Validation

- The `mondayDate` parameter **MUST be a Monday**
- Format must be `yyyy-MM-dd` (e.g., "2024-12-30")
- Invalid dates will return 400 error

Query Parameters

- All parameters are passed in the URL as query strings
- `staffId` and `mondayDate` are **required**
- `isSpecialPayRate` is **optional** (default: `false`)

Pay Rate Options

- **isSpecialPayRate = false** (default): Uses staff's current **StandardPayRate** from database
- **isSpecialPayRate = true**: Uses default Horticulture Award 2025 rates based on staff's Role and ContractType
- Can be omitted entirely (will default to **false**)

🌐 Browser Accessible

- Since it's a GET endpoint, you can **test directly in a browser**
- Just paste the URL with query parameters
- No need for Postman or cURL for basic testing

🕒 Hours Calculation

- Automatically rounds to nearest 5 minutes (Award compliance)
- Deducts 30 min unpaid break if shift > 5 hours
- Calculates from clock-in/out events stored in Event table

📊 Tax Calculation

- Based on ATO 2024-25 resident tax tables
- Medicare levy is **NOT** included (as per requirements)
- Projected annual income used for tax bracket determination

Integration Examples

React Component Example

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

function PayrollCalculator() {
  const [payroll, setPayroll] = useState(null);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState(null);

  const calculatePayroll = async (staffId, mondayDate, useSpecialRate = false) =>
  {
    setLoading(true);
    setError(null);

    try {
      const response = await axios.get('/api/payroll/calculate', {
        params: {
          staffId,
          mondayDate,
          isSpecialPayRate: useSpecialRate
        }
      });
    }

    setPayroll(response.data);
  }
}
```

```
    } catch (err) {
      setError(err.response?.data?.message || 'Error calculating payroll');
    } finally {
      setLoading(false);
    }
  };

  return (
    <div>
      <button onClick={() => calculatePayroll(1, '2024-12-30', false)}>
        Calculate with Current Rate
      </button>
      <button onClick={() => calculatePayroll(1, '2024-12-30', true)}>
        Calculate with Award Rate
      </button>

      {loading && <p>Calculating...</p>}
      {error && <p className="error">{error}</p>}
      {payroll && (
        <div className="payroll-summary">
          <h3>{payroll.staffName}</h3>
          <p>Hours Worked: {payroll.totalHoursWorked}</p>
          <p>Gross Pay: ${payroll.grossWeeklyPay.toFixed(2)}</p>
          <p>Tax: ${payroll.weeklyPAYG.toFixed(2)}</p>
          <p>Net Pay: ${payroll.netPay.toFixed(2)}</p>
          <p>Super: ${payroll.employerSuperannuation.toFixed(2)}</p>
        </div>
      )}
    </div>
  );
}
```

Troubleshooting

Common Errors and Solutions

Error	Cause	Solution
"The provided date is not a Monday"	Date is not Monday	Use a Monday date (check calendar)
"Invalid date format"	Wrong date format	Use yyyy-MM-dd format
"Staff with ID X not found"	Staff doesn't exist	Verify staff ID exists in database
Gross pay is \$0	No clock events for week	Ensure staff has clock-in/out events

Validation Checklist

- ☒ Staff ID is greater than 0
- ☒ Monday date parameter is provided
- ☒ Date is a Monday

- ☒ Date format is `yyyy-MM-dd`
- ☒ Staff ID exists in database
- ☒ Staff has Role and ContractType set (for special rates)
- ☒ Staff has clock-in/out events for the week

Quick Test

Copy and paste this URL in your browser to test:

```
http://localhost:5000/api/payroll/calculate?staffId=1&mondayDate=2024-12-30
```

If it works, you should see JSON response immediately!

Related APIs

- **Pay Rates API:** `/api/payrates` - Manage staff pay rate configurations
 - **Events API:** `/api/events` - Clock-in/out event management
 - **Staff API:** `/api/staffs` - Staff management
-

Support

For issues or questions:

- Check this documentation first
 - Review the Feature 2 Payroll Calculation requirements document
 - Contact the backend development team
-

Last Updated: January 2025

API Version: 1.0

Compliance: Horticulture Award 2025, ATO Tax Tables 2024-25