Saifee Hospital - Insurance Integration API Documentation

Table of Contents

Overview

This document provides comprehensive API specifications for integrating insurance providers with Saifee Hospital Management System (SHAMS). The API enables seamless exchange of patient information, pre-authorization requests, claim submissions, and status tracking.

Base URL

```
{{BASE_URL}}/api/v1
```

Content Type

All requests and responses use `application/json` unless specified otherwise.

Authentication

1. Get Authentication Token

Endpoint: `POST /Token`

Description: Obtains an access token required for all subsequent API calls.

Request Format: `multipart/form-data`

Request Parameters:

Parameter	Type	Required	Description
username	string	Yes	Provider username credentials
password	string	Yes	Provider password credentials
providerid	string	Yes	Unique provider identifier

Request Example:

```
http
```

```
POST /Token HTTP/1.1
Content-Type: multipart/form-data

username=provider_username
password=provider_password
providerid=12345678
```

Response:

```
json

{
 "Status": "Success",
 "Description": {
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 3600
 }
}
```

Response Fields:

Field	Type	Description
Status	string	Success/Error indicator
Description.access_token	string	JWT token for authentication
Description.token_type	string	Type of token (Bearer)
Description.expires_in	integer	Token expiration time in seconds

---

API Endpoints

2. Get Card Details

Endpoint: `GET /Getcarddetails`

Description: Retrieves member/patient insurance card details and eligibility information.

Authentication: Bearer Token (Required)

Query Parameters:

Parameter	Type	Required	Description
MemberNo	string	Yes	Member/Card number

Request Example:

```
http
```

```
GET /Getcarddetails?MemberNo=11651352 HTTP/1.1
Authorization: Bearer {access_token}
```

Response:

```
json

{
 "Status": "Success",
 "Description": {
  "MemberNo": "11651352",
  "MemberName": "John Doe",
  "DateOfBirth": "1990-05-15",
  "Gender": "M",
  "Phone": "0712345678",
  "LastAuthorization": "2024-01-15",
  "PolicyStatus": "Active",
  "CoverageDetails": {
   "InpatientLimit": "5000000",
   "OutpatientLimit": "500000",
   "Copay": "10"
  }
 }
}
```

Response Fields:

Field	Type	Description
Status	string	Request status (Success/Error)
MemberNo	string	Member card number
MemberName	string	Full name of the member
DateOfBirth	string	Date of birth (YYYY-MM-DD)
Gender	string	Gender (M/F)
Phone	string	Contact phone number
LastAuthorization	string	Last authorization date
PolicyStatus	string	Current policy status
CoverageDetails	object	Coverage limits and copay info

---

3. Get Price List

Endpoint: `GET /GetPriceList`

Description: Retrieves the agreed-upon pricing list for medical services and items.

Authentication: Bearer Token (Required)

Request Example:

```http
GET /GetPriceList HTTP/1.1
Authorization: Bearer {access_token}
```

Response:

```json
{
 "Status": "Success",
 "Description": [
  {
   "ItemCode": "64933673",
   "ItemName": "Consultation - General Practitioner",
   "ItemPrice": "50000.00",
   "ProviderID": "12079538",
   "Category": "Consultation"
  },
  {
   "ItemCode": "64925043",
   "ItemName": "Complete Blood Count (CBC)",
   "ItemPrice": "25000.00",
   "ProviderID": "12079538",
   "Category": "Laboratory"
  }
 ]
}
```

Response Fields:

Field	Type	Description
Status	string	Request status
Description	array	Array of price items
ItemCode	string	Unique item identifier
ItemName	string	Item/service description
ItemPrice	decimal	Agreed price
ProviderID	string	Provider identifier
Category	string	Service category

4. Get Procedure List

Endpoint: GET /GetProcedureList

Description: Retrieves the list of medical procedures covered under the insurance plan.

Authentication: Bearer Token (Required)

Request Example:

```http
GET /GetProcedureList HTTP/1.1
Authorization: Bearer {access_token}
```

Response:

```json
{
 "Status": "Success",
 "Description": [
  {
   "ProcedureCode": "JIC0333",
   "ProcedureName": "Minor Surgical Procedure",
   "BenefitCode": "7905",
   "RequiresPreAuth": true,
   "MaxAmount": "500000.00"
  }
 ]
}
```

---

5. Check Verification

Endpoint: GET /CheckVerification

Description: Verifies member eligibility for a specific visit date.

Authentication: Bearer Token (Required)

Query Parameters:

Parameter	Type	Required	Description
MemberNo	string	Yes	Member card number
VisitDate	string	Yes	Visit date (YYYY-MM-DD)

Request Example:

```http
```

```
GET /CheckVerification?MemberNo=12196895&VisitDate=2025-10-10 HTTP/1.1
Authorization: Bearer {access_token}
```

Response:

```json
{
  "Status": "Success",
  "Description": {
    "IsEligible": true,
    "MemberNo": "12196895",
    "VisitDate": "2025-10-10",
    "RemainingLimit": "450000.00",
    "Message": "Member is eligible for services"
  }
}
```

---

6. Verify Items

Endpoint: `POST /VerifyItems`

Description: Verifies specific items/services before providing treatment.

Authentication: Bearer Token (Required)

Content-Type: `application/x-www-form-urlencoded`

Request Parameters:

Parameter	Type	Required	Description
BenefitCode	string	Yes	Benefit code for the service
ProcedureId	string	Yes	Procedure identifier
MemberNo	string	Yes	Member card number
Amount	decimal	Yes	Total amount to verify
VerifyItems	JSON array	Yes	Array of items to verify
PreauthSubmissionID	string	No	Pre-auth submission ID (if exists)
VisitDate	string	Yes	Visit date (YYYY-MM-DD)

VerifyItems Structure:

```json
```

```
[
  {
    "ItemId": "64933673",
    "ItemQuantity": "1",
    "ItemPrice": "15000.00"
  }
]
```

Request Example:

http

```
POST /VerifyItems HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/x-www-form-urlencoded

BenefitCode=7905&ProcedureId=JIC1275&MemberNo=12182196&Amount=15000&VerifyItems=[{"ItemId":"64933673","
```

Response:

json

```
{
  "Status": "Success",
  "Description": {
    "VerificationID": "VER123456",
    "ApprovedAmount": "15000.00",
    "Items": [
      {
        "ItemId": "64933673",
        "Status": "Approved",
        "ApprovedQuantity": "1",
        "ApprovedAmount": "15000.00"
      }
    ]
  }
}
```

---

7. Send Pre-authorization

Endpoint: POST /SendPreauthorization

Description: Submits a pre-authorization request for planned treatment.

Authentication: Bearer Token (Required)

Content-Type: application/json

Request Body:

json

```
{
 "entities": [
  {
   "ClaimYear": "2025",
   "ClaimMonth": "6",
   "CardNo": "12197801",
   "FirstName": "John",
   "LastName": "Doe",
   "Gender": "M",
   "DateOfBirth": "1986-11-28",
   "Age": "38",
   "TelephoneNo": "0746333328",
   "PatientFileNo": "68138",
   "AuthorizationNo": "278588791",
   "AttendanceDate": "2025-06-10",
   "PatientTypeCode": "OUT",
   "DateAdmitted": "",
   "DateDischarged": "",
   "PractitionerNo": "MCTOL67154",
   "CreatedBy": "Dr. John Smith",
   "DateCreated": "2025-06-10",
   "LastModifiedBy": null,
   "LastModified": null,
   "FolioDiseases": [
    {
     "DiseaseCode": "J32.9",
     "Remarks": null,
     "Status": "Provisional",
     "CreatedBy": "Dr. John Smith",
     "DateCreated": "2025-06-10 14:44:47",
     "LastModifiedBy": null,
     "LastModified": null
    },
    {
     "DiseaseCode": "J32.9",
     "Remarks": null,
     "Status": "Final",
     "CreatedBy": "Dr. John Smith",
     "DateCreated": "2025-06-10 14:44:47",
     "LastModifiedBy": null,
     "LastModified": null
    }
   ],
   "FolioItems": [
    {
     "ItemCode": "62162150",
```

```
        "ItemQuantity": 1,
        "UnitPrice": "330000",
        "AmountClaimed": "330000.00",
        "CreatedBy": "Dr. John Smith",
        "DateCreated": "2025-06-10",
        "LastModifiedBy": null,
        "LastModified": null
      }
    ],
    "QualificationID": null,
    "AmountClaimed": "390000.00",
    "jubileeProcedure": "JIC0333",
    "jubileeBenefits": "7905",
    "BillNo": "2231131",
    "ProviderID": "12079538"
  }
 ]
}
```


Request Fields:

Field	Type	Required	Description
ClaimYear	string	Yes	Year of claim (YYYY)
ClaimMonth	string	Yes	Month of claim (MM)
CardNo	string	Yes	Member card number
FirstName	string	Yes	Patient first name
LastName	string	Yes	Patient last name
Gender	string	Yes	Gender (M/F)
DateOfBirth	string	Yes	Date of birth (YYYY-MM-DD)
Age	string	Yes	Patient age
TelephoneNo	string	Yes	Contact number
PatientFileNo	string	Yes	Hospital file number
AuthorizationNo	string	Yes	Authorization number
AttendanceDate	string	Yes	Date of attendance (YYYY-MM-DD)
PatientTypeCode	string	Yes	OUT (Outpatient) / IN (Inpatient)
DateAdmitted	string	Conditional	Required for inpatients (YYYY-MM-DD)
DateDischarged	string	Conditional	Required for inpatients (YYYY-MM-DD)
PractitionerNo	string	Yes	Practitioner license number
CreatedBy	string	Yes	User who created the record
DateCreated	string	Yes	Creation date (YYYY-MM-DD)
FolioDiseases	array	Yes	Array of diagnoses
FolioItems	array	Yes	Array of claimed items
AmountClaimed	decimal	Yes	Total amount claimed
jubileeProcedure	string	Yes	Procedure code (insurance-specific)
jubileeBenefits	string	Yes	Benefit code (insurance-specific)
BillNo	string	Yes	Hospital bill number
ProviderID	string	Yes	Provider identifier

FolioDiseases Structure:

Field	Type	Required	Description
DiseaseCode	string	Yes	ICD-10 diagnosis code
Remarks	string	No	Additional notes
Status	string	Yes	Provisional/Final
CreatedBy	string	Yes	Practitioner name
DateCreated	string	Yes	Date created (YYYY-MM-DD HH:MM:SS)

FolioItems Structure:

Field	Type	Required	Description
ItemCode	string	Yes	Item/service code
ItemQuantity	integer	Yes	Quantity claimed
UnitPrice	decimal	Yes	Price per unit
AmountClaimed	decimal	Yes	Total amount for item
CreatedBy	string	Yes	User who added item
DateCreated	string	Yes	Date created (YYYY-MM-DD)

Response:

```json
{
 "Status": "Success",
 "Description": {
  "SubmissionID": "1291008",
  "AuthorizationNo": "278588791",
  "Status": "Pending",
  "Message": "Pre-authorization submitted successfully"
 }
}
```

---

8. Update Pre-authorization

Endpoint: `POST /UpdatePreauthorization`

Description: Updates an existing pre-authorization request.

Authentication: Bearer Token (Required)

Content-Type: `application/json`

Request Body: Same structure as SendPreauthorization

Note: Include the original SubmissionID in the request for tracking purposes.

Response:

```json
```

```
{
 "Status": "Success",
 "Description": {
  "SubmissionID": "1291008",
  "Status": "Updated",
  "Message": "Pre-authorization updated successfully"
 }
}
```


---

9. Send Claim

Endpoint: POST /SendClaim

Description: Submits a final claim after service delivery.

Authentication: Bearer Token (Required)

Content-Type: application/json

Request Body:

```
json
```

```
{
 "entities": [
  {
   "FolioID": null,
   "ClaimYear": "2023",
   "ClaimMonth": "12",
   "FolioNo": "1",
   "SerialNo": "07141\\12\\2023\\1240",
   "CardNo": "11768722",
   "FirstName": "JOHN",
   "LastName": "DOE",
   "Gender": "Male",
   "DateOfBirth": "1988-01-15",
   "Age": "35",
   "TelephoneNo": "714281313",
   "PatientFileNo": "1 12 23",
   "PatientFile": "Base64EncodedPDFContent...",
   "AuthorizationNo": "161041573",
   "AttendanceDate": "2023-12-11",
   "PatientTypeCode": "OUT",
   "DateAdmitted": null,
   "DateDischarged": null,
   "PractitionerNo": "12345",
   "CreatedBy": "Dr. Jane Smith",
   "DateCreated": "2023-12-11",
   "LastModifiedBy": null,
   "LastModified": null,
   "FolioDiseases": [
    {
     "DiseaseCode": "O28.9",
     "Remarks": null,
     "Status": "Provisional",
     "CreatedBy": "Dr. Jane Smith",
     "DateCreated": "2023-12-11 09:06:04",
     "LastModifiedBy": null,
     "LastModified": null
    },
    {
     "DiseaseCode": "O28.9",
     "Remarks": null,
     "Status": "Final",
     "CreatedBy": "Dr. Jane Smith",
     "DateCreated": "2023-12-11 09:06:04",
     "LastModifiedBy": null,
     "LastModified": null
    }
```

```
    ],
    "FolioItems": [
      {
        "ItemCode": "64925043",
        "OtherDetails": null,
        "ItemQuantity": 1,
        "UnitPrice": "20000.00",
        "AmountClaimed": "20000.00",
        "ApprovalRefNo": null,
        "CreatedBy": "Dr. Jane Smith",
        "DateCreated": "2023-12-11 09:06:04",
        "LastModifiedBy": null,
        "LastModified": null
      }
    ],
    "ClaimFile": "Base64EncodedClaimDocumentPDF...",
    "FacilityCode": null,
    "ClinicalNotes": "<HTML formatted clinical notes>",
    "DelayReason": null,
    "BillNo": "1240",
    "QualificationID": null,
    "SubmissionChannel": null,
    "EmergencyAuthorizationReason": null,
    "LateAuthorizationReason": null,
    "LateSubmissionReason": null,
    "AmountClaimed": "32875",
    "ProviderID": "11767036"
  }
 ]
}
```


Additional Fields (specific to claims):

Field	Type	Required	Description
FolioID	string	No	Internal folio identifier
FolioNo	string	Yes	Folio number
SerialNo	string	Yes	Unique serial number
PatientFile	string	Yes	Base64 encoded patient file (PDF)
ClaimFile	string	Yes	Base64 encoded claim document (PDF)
ClinicalNotes	string	Yes	HTML formatted clinical notes
FacilityCode	string	No	Facility code if applicable
DelayReason	string	Conditional	Required if delayed submission
SubmissionChannel	string	No	Channel of submission
EmergencyAuthorizationReason	string	Conditional	Required for emergency cases
LateAuthorizationReason	string	Conditional	Required if authorization was late
LateSubmissionReason	string	Conditional	Required if submission is late

Response:

```json
{
  "Status": "Success",
  "Description": {
    "ClaimID": "CLM123456",
    "SubmissionDate": "2023-12-11",
    "Status": "Received",
    "Message": "Claim submitted successfully"
  }
}
```

---

10. Get Pre-authorization Status

Endpoint: `GET /getPreauthorizationStatus`

Description: Retrieves the status of a submitted pre-authorization request.

Authentication: Bearer Token (Required)

Query Parameters:

Parameter	Type	Required	Description
submissionID	string	Yes	Pre-authorization submission ID

Request Example:

```http
```

```
GET /getPreauthorizationStatus?submissionID=1291008 HTTP/1.1
Authorization: Bearer {access_token}
```

Response:

```json
{
  "Status": "Success",
  "Description": {
    "SubmissionID": "1291008",
    "AuthorizationNo": "278588791",
    "Status": "Approved",
    "ApprovedAmount": "350000.00",
    "RejectionReason": null,
    "ReviewComments": "Approved for procedure",
    "ReviewDate": "2025-06-11"
  }
}
```

Status Values:

- `Pending`: Under review
- `Approved`: Pre-authorization approved
- `Rejected`: Pre-authorization rejected
- `MoreInfoRequired`: Additional information needed
- `Cancelled`: Pre-authorization cancelled

---

11. Get Admission Status

Endpoint: `GET /getAdmissionStatus.php`

Description: Retrieves admission authorization status and details.

Authentication: Bearer Token (Required)

Query Parameters:

Parameter	Type	Required	Description
authorizationNo	string	Yes	Authorization number

Request Example:

```http
```

```
GET /getAdmissionStatus.php?authorizationNo=325074175 HTTP/1.1
Authorization: Bearer {access_token}
```

Response:

json


```
{
  "Status": "Success",
  "Description": {
    "authorizationNo": "325074175",
    "memberNo": "12345678",
    "fromDate": "2025-06-10",
    "toDate": "2025-06-15",
    "approvedAmount": "2500000.00",
    "usedAmount": "1200000.00",
    "remainingAmount": "1300000.00",
    "admissionStatus": "Active"
  }
}
```


Response Fields:

Field	Type	Description
authorizationNo	string	Authorization number
memberNo	string	Member card number
fromDate	string	Admission start date
toDate	string	Admission end date
approvedAmount	decimal	Total approved amount
usedAmount	decimal	Amount used so far
remainingAmount	decimal	Remaining available amount
admissionStatus	string	Current admission status

Data Models

Common Data Types

Patient Types

- `OUT`: Outpatient
- `IN`: Inpatient
- `EME`: Emergency

Diagnosis Status

- `Provisional`: Initial/working diagnosis
- `Final`: Confirmed diagnosis

Gender Codes

- `M`: Male
- `F`: Female

Date Formats

- Date: `YYYY-MM-DD`
- DateTime: `YYYY-MM-DD HH:MM:SS`

Response Codes

HTTP Status Codes

Code	Status	Description
200	OK	Request successful
201	Created	Resource created successfully
400	Bad Request	Invalid request parameters
401	Unauthorized	Invalid or expired token
403	Forbidden	Insufficient permissions
404	Not Found	Resource not found
422	Unprocessable Entity	Validation error
500	Internal Server Error	Server error

Application Status Codes

All API responses include a `Status` field:

Status	Description
Success	Request processed successfully
Error	Request failed, check error message
Warning	Request processed with warnings

Error Response Format:

```
json
```

```
{
  "Status": "Error",
  "Description": {
    "ErrorCode": "ERR_001",
    "ErrorMessage": "Member not found",
    "Details": "The specified member number does not exist in the system"
  }
}
```


Common Error Codes

Error Code	Description
ERR_001	Member not found
ERR_002	Invalid authentication credentials
ERR_003	Token expired
ERR_004	Insufficient coverage
ERR_005	Service not covered
ERR_006	Pre-authorization required
ERR_007	Invalid item code
ERR_008	Duplicate submission
ERR_009	Missing required field
ERR_010	Invalid date format

Implementation Guide

Step 1: Authentication Flow

1. Obtain provider credentials (username, password, providerid)
2. Call `/Token` endpoint with credentials
3. Store the access token securely
4. Include token in all subsequent requests as Bearer token
5. Monitor token expiration and refresh as needed

Step 2: Patient Verification

1. Collect patient's member card number
2. Call `/Getcarddetails` to verify eligibility
3. Validate coverage and limits
4. Call `/CheckVerification` for visit date verification

Step 3: Service Verification

1. Select services/items to be provided
2. Call `/VerifyItems` for coverage verification
3. Review approved items and amounts
4. Proceed with treatment only for approved items

Step 4: Pre-authorization (for high-cost/planned procedures)

1. Prepare diagnosis and treatment plan
2. Compile list of services/items
3. Submit `/SendPreauthorization` request
4. Monitor status using `/getPreauthorizationStatus`
5. Wait for approval before proceeding
6. If changes needed, use `/UpdatePreauthorization`

Step 5: Claim Submission

1. Complete patient treatment
2. Generate clinical notes and supporting documents
3. Convert documents to Base64 format
4. Compile all diagnosis codes (Provisional and Final)
5. Compile all items provided
6. Submit `/SendClaim` with complete documentation
7. Store claim ID for tracking

Step 6: Status Monitoring

1. Regularly check pre-authorization status
2. Monitor admission status for inpatients
3. Track claim processing status
4. Handle rejections and requests for more information

Best Practices

1. **Security**
   - Store credentials securely
   - Use HTTPS for all communications
   - Implement token refresh mechanism

- Never log sensitive patient data

2. **Data Validation**
   - Validate all data before submission
   - Ensure date formats are correct
   - Verify ICD-10 codes are valid
   - Check item codes against price list
3. **Error Handling**
   - Implement retry logic for network failures
   - Log all API interactions
   - Provide clear error messages to users
   - Handle token expiration gracefully
4. **Performance**
   - Implement caching for price lists
   - Batch requests where possible
   - Use connection pooling
   - Monitor API response times
5. **Compliance**
   - Maintain audit trails
   - Ensure data privacy compliance
   - Follow insurance provider guidelines
   - Keep documentation updated

Appendix

File Encoding Guidelines

When submitting documents (PatientFile, ClaimFile), follow these guidelines:

1. **Supported Formats**: PDF
2. **Encoding**: Base64
3. **Maximum Size**: 5MB per file
4. **Required Documents**:
   - Patient medical file

- Claim supporting documents
- Clinical notes (HTML format)

ICD-10 Code Guidelines

- Use standard ICD-10 codes
- Include both Provisional and Final diagnoses
- Provide codes for primary and secondary conditions
- Format: `XX.XX` (e.g., `J32.9`)