

Compliance & Reporting Service

API Contract Specification v1.0

Document Information

- **Service Name:** Compliance & Reporting Service
 - **API Version:** 1.0.0
 - **Document Date:** September 24, 2025
 - **Document Owner:** Shireen Hussain J
 - **Classification:** Internal Use Only
 - **Technology Stack:** Java 17 + Spring Boot + Apache POI
-

1. Service Overview

1.1 Purpose

The Compliance & Reporting Service manages the generation, validation, and submission of ESG reports across multiple regulatory standards. It serves as the central engine for transforming raw ESG data into compliant, audit-ready reports for various stakeholders and regulatory bodies.

1.2 Core Capabilities

- **Multi-Standard Reporting:** CSRD, SFDR, ISSB, GRI, CDP, GHG Protocol compliance
- **Template Management:** Dynamic report templates with regulatory updates
- **Report Generation:** Automated creation of PDF, Word, Excel, and XML formats
- **Approval Workflows:** Multi-level review and approval processes
- **Submission Management:** Direct integration with regulatory portals
- **Audit Trail:** Complete documentation and version control
- **Validation Engine:** Compliance rule checking and gap analysis

1.3 Base URLs

- **Production:** <https://api.sweep.com/v1/compliance-reporting>
- **Staging:** <https://api-staging.sweep.com/v1/compliance-reporting>
- **Development:** <https://api-dev.sweep.com/v1/compliance-reporting>

1.4 Authentication

- **Primary:** OAuth 2.0 Bearer Token (JWT format)

- **Service-to-Service:** API Key authentication
 - **Required Roles:** ESG_ANALYST, SUSTAINABILITY_MANAGER, COMPLIANCE_OFFICER, AUDITOR
-

2. API Endpoints

2.1 Health Check

GET /health

Purpose: Returns service health and regulatory framework status

Response (200 OK):

json

```
{
  "status": "healthy",
  "timestamp": "2025-09-24T15:45:00Z",
  "version": "1.0.0",
  "uptime": "120h 45m 30s",
  "regulatoryFrameworks": {
    "CSRD": {
      "status": "active",
      "version": "2024.1",
      "lastUpdate": "2025-09-01T00:00:00Z",
      "templatesLoaded": 12
    },
    "SFDR": {
      "status": "active",
      "version": "2024.2",
      "pailIndicators": 18,
      "templatesLoaded": 8
    },
    "GRI": {
      "status": "active",
      "version": "2023",
      "universalStandards": 3,
      "topicStandards": 33
    }
  },
  "reportQueue": {
    "pendingReports": 15,
    "inProgress": 8,
    "averageProcessingTime": "12m 30s"
  }
}
```

2.2 Report Generation

POST /reports/generate

Purpose: Generate compliance reports based on specified standards and parameters

Request Body:

json

```
{
  "reportJobId": "report-job-abc123",
  "organizationId": "org-123",
  "reportConfig": {
    "standard": "CSRD",
    "template": "ESRS_E1_CLIMATE_CHANGE",
    "reportingPeriod": {
      "startDate": "2024-01-01T00:00:00Z",
      "endDate": "2024-12-31T23:59:59Z",
      "fiscalYear": "2024"
    },
    "scope": {
      "entities": ["entity-dubai-hq", "entity-abu-dhabi-plant"],
      "geographies": ["UAE", "KSA"],
      "businessUnits": ["manufacturing", "operations"]
    },
    "outputFormats": ["PDF", "XLSX", "XML"],
    "language": "EN",
    "currency": "USD"
  },
  "dataFilters": {
    "includeEstimatedData": true,
    "qualityThreshold": 85.0,
    "includeThirdPartyData": true,
    "dataAssurance": "LIMITED"
  },
  "customizations": {
    "includeBranding": true,
    "executiveSummary": true,
    "detailedBreakdowns": true,
    "comparativePeriods": ["2023", "2022"]
  }
}
```

Response (202 Accepted):

json

```
{  
  "reportJobId": "report-job-abc123",  
  "status": "PROCESSING",  
  "standard": "CSRD",  
  "template": "ESRS_E1_CLIMATE_CHANGE",  
  "estimatedCompletionTime": "15-25 minutes",  
  "dataPointsToProcess": 1247,  
  "calculationsRequired": 45,  
  "statusUrl": "/reports/generate/status/report-job-abc123",  
  "submittedAt": "2025-09-24T16:00:00Z"  
}
```

GET /reports/generate/status/{jobId}

Purpose: Check report generation status and retrieve completed reports

Response (200 OK):

json

```
{
  "reportJobId": "report-job-abc123",
  "status": "COMPLETED",
  "progress": {
    "dataCollection": "COMPLETED",
    "calculations": "COMPLETED",
    "validation": "COMPLETED",
    "formatting": "COMPLETED",
    "percentComplete": 100
  },
  "results": {
    "reportsGenerated": 3,
    "totalPages": 156,
    "dataPointsProcessed": 1247,
    "calculationsPerformed": 45,
    "validationIssues": 2,
    "outputs": [
      {
        "format": "PDF",
        "fileSize": "12.4MB",
        "downloadUrl": "/reports/download/report-job-abc123.pdf",
        "expiresAt": "2025-09-31T16:00:00Z"
      },
      {
        "format": "XLSX",
        "fileSize": "3.8MB",
        "downloadUrl": "/reports/download/report-job-abc123.xlsx",
        "worksheets": 12
      }
    ]
  },
  "compliance": {
    "standardCompliance": "FULLY_COMPLIANT",
    "mandatoryDisclosures": {
      "total": 85,
      "completed": 83,
      "missing": 2,
      "completenessPercentage": 97.6
    }
  },
  "validationResults": [
    {
      "ruleId": "ESRS-E1-6",
      "status": "WARNING",
      "message": "Energy consumption data missing for Q4 renewable sources",
      "impact": "LOW",
      "recommendation": "Include estimated renewable energy consumption"
```

```
}  
]  
,  
"completedAt": "2025-09-24T16:18:45Z"  
}
```

2.3 Report Templates Management

GET /templates

Purpose: List available report templates by standard

Query Parameters:

- `standard`: Filter by regulatory standard (CSRD, SFDR, ISSB, GRI, CDP, GHG_PROTOCOL)
- `category`: Filter by category (ENVIRONMENTAL, SOCIAL, GOVERNANCE)
- `language`: Template language (EN, FR, DE, ES)
- `version`: Template version

Response (200 OK):

json

```

{
  "templates": [
    {
      "templateId": "CSRD_ESRS_E1",
      "name": "CSRD - Climate Change (ESRS E1)",
      "standard": "CSRD",
      "category": "ENVIRONMENTAL",
      "version": "1.2.0",
      "description": "CSRD compliance template for ESRS E1 Climate Change disclosure",
      "languages": ["EN", "FR", "DE"],
      "requiredDataPoints": [
        "scope_1_emissions",
        "scope_2_emissions",
        "scope_3_emissions",
        "energy_consumption",
        "renewable_energy_percentage"
      ],
      "mandatoryDisclosures": 15,
      "optionalDisclosures": 8,
      "estimatedPages": 25,
      "lastUpdated": "2025-08-15T10:00:00Z",
      "regulatoryDeadline": "2026-03-31T00:00:00Z"
    },
    {
      "templateId": "GRI_302_ENERGY",
      "name": "GRI 302: Energy 2016",
      "standard": "GRI",
      "category": "ENVIRONMENTAL",
      "version": "2016.1",
      "description": "GRI Standard for energy consumption and intensity reporting",
      "requiredDataPoints": [
        "energy_consumption_within_organization",
        "energy_consumption_outside_organization",
        "energy_intensity"
      ],
      "disclosures": [
        "302-1", "302-2", "302-3", "302-4", "302-5"
      ]
    }
  ],
  "totalTemplates": 45,
  "standardsSupported": 6,
  "lastFrameworkUpdate": "2025-09-01T00:00:00Z"
}

```

GET /templates/{templateId}

Purpose: Get detailed template information and requirements

Response (200 OK):

json

```
{
  "templateId": "CSRD_ESRS_E1",
  "name": "CSRD - Climate Change (ESRS E1)",
  "standard": "CSRD",
  "version": "1.2.0",
  "structure": {
    "sections": [
      {
        "sectionId": "E1-1",
        "title": "Transition plan for climate change mitigation",
        "mandatory": true,
        "subsections": [
          {
            "subsectionId": "E1-1.1",
            "title": "Transition plan",
            "dataRequirements": [
              "decarbonization_targets",
              "transition_timeline",
              "investment_plans"
            ],
            "validationRules": [
              {
                "ruleId": "E1-1.1-R1",
                "description": "Transition plan must include specific targets",
                "severity": "MANDATORY"
              }
            ]
          }
        ]
      }
    ]
  },
  "dataMapping": {
    "scope_1_emissions": {
      "metricTypes": ["SCOPE_1_EMISSIONS"],
      "aggregation": "SUM",
      "units": ["kg CO2e", "tonnes CO2e"],
      "timeGranularity": "ANNUAL"
    }
  },
  "outputOptions": {
    "formats": ["PDF", "DOCX", "XLSX", "XML"],
    "customBranding": true,
    "multiLanguage": true,
    "digitalSignature": true
  }
}
```

```
}  
}
```

2.4 Compliance Validation

POST /validate/compliance

Purpose: Validate data completeness and compliance before report generation

Request Body:

```
json  
  
{  
  "validationJobId": "validation-job-456def",  
  "organizationId": "org-123",  
  "validationConfig": {  
    "standard": "CSRD",  
    "template": "ESRS_E1_CLIMATE_CHANGE",  
    "reportingPeriod": {  
      "startDate": "2024-01-01T00:00:00Z",  
      "endDate": "2024-12-31T23:59:59Z"  
    },  
    "strictMode": true,  
    "includeWarnings": true  
  },  
  "scope": {  
    "entities": ["entity-dubai-hq", "entity-abu-dhabi-plant"],  
    "dataTypes": ["EMISSIONS", "ENERGY", "WASTE", "WATER"]  
  }  
}
```

Response (200 OK):

```
json
```

```
{
  "validationJobId": "validation-job-456def",
  "status": "COMPLETED",
  "overallCompliance": {
    "status": "PARTIALLY_COMPLIANT",
    "score": 87.3,
    "readinessLevel": "NEEDS_IMPROVEMENT"
  },
  "mandatoryRequirements": {
    "total": 15,
    "satisfied": 13,
    "missing": 2,
    "complianceRate": 86.7
  },
  "validationResults": [
    {
      "requirementId": "ESRS-E1-1",
      "title": "Transition plan for climate change mitigation",
      "status": "COMPLIANT",
      "completeness": 100,
      "dataQuality": 95.2
    },
    {
      "requirementId": "ESRS-E1-5",
      "title": "Energy consumption and mix",
      "status": "NON_COMPLIANT",
      "completeness": 65,
      "issues": [
        {
          "type": "MISSING_DATA",
          "severity": "CRITICAL",
          "description": "Renewable energy percentage missing for Q3 and Q4",
          "dataPoints": ["renewable_energy_q3", "renewable_energy_q4"],
          "recommendation": "Collect or estimate missing renewable energy data"
        }
      ]
    }
  ],
  "gapAnalysis": {
    "criticalGaps": 2,
    "warnings": 5,
    "recommendations": [
      "Implement automated renewable energy tracking",
      "Establish quarterly data validation procedures",
      "Consider third-party data assurance for material metrics"
    ]
  },
}
```

```
"estimatedEffortToCompliance": "40-60 hours"
},
"completedAt": "2025-09-24T16:25:00Z"
}
```

2.5 Approval Workflows

POST /approvals/initiate

Purpose: Start approval workflow for generated reports

Request Body:

json

```
{
  "approvalJobId": "approval-job-789ghi",
  "reportJobId": "report-job-abc123",
  "organizationId": "org-123",
  "workflowConfig": {
    "workflowType": "CSRD_APPROVAL",
    "approvers": [
      {
        "level": 1,
        "role": "ESG_ANALYST",
        "users": ["user-sarah-chen"],
        "required": true,
        "deadline": "2025-09-26T17:00:00Z"
      },
      {
        "level": 2,
        "role": "SUSTAINABILITY_MANAGER",
        "users": ["user-michael-brown"],
        "required": true,
        "deadline": "2025-09-28T17:00:00Z"
      },
      {
        "level": 3,
        "role": "COMPLIANCE_OFFICER",
        "users": ["user-lisa-wilson", "user-david-kumar"],
        "required": false,
        "parallelApproval": true
      }
    ],
    "escalationRules": {
      "autoEscalateAfter": "24h",
      "escalateTo": "user-ceo-office"
    },
    "notifications": {
      "email": true,
      "inApp": true,
      "webhook": "https://client.sweep.com/webhooks/approval-status"
    }
  }
}
```

Response (202 Accepted):

json

```
{
  "approvalJobId": "approval-job-789ghi",
  "workflowId": "workflow-123abc",
  "status": "INITIATED",
  "currentLevel": 1,
  "currentApprovers": ["user-sarah-chen"],
  "overallDeadline": "2025-09-30T17:00:00Z",
  "statusUrl": "/approvals/status/approval-job-789ghi",
  "initiatedAt": "2025-09-24T16:30:00Z"
}
```

POST /approvals/{approvalJobId}/action

Purpose: Approve, reject, or request changes to reports

Request Body:

```
json

{
  "action": "APPROVE",
  "approver": "user-sarah-chen",
  "level": 1,
  "comments": "Report looks comprehensive and meets CSRD requirements",
  "conditions": [],
  "attachments": [
    {
      "name": "approval_certificate.pdf",
      "url": "/attachments/approval_certificate_789ghi.pdf"
    }
  ]
}
```

Response (200 OK):

```
json
```

```
{
  "approvalJobId": "approval-job-789ghi",
  "actionResult": "APPROVED_LEVEL_1",
  "nextLevel": 2,
  "nextApprovers": ["user-michael-brown"],
  "workflowStatus": "IN_PROGRESS",
  "actionRecorded": "2025-09-24T17:15:00Z",
  "notifications": {
    "sent": ["user-michael-brown"],
    "method": ["EMAIL", "IN_APP"]
  }
}
```

2.6 Regulatory Submission

POST /submit/regulatory

Purpose: Submit approved reports to regulatory portals

Request Body:

json


```
{
  "submissionJobId": "submission-job-abc456",
  "reportJobId": "report-job-abc123",
  "organizationId": "org-123",
  "submissionConfig": {
    "standard": "CSRD",
    "regulatoryPortal": "ESMA_SINGLE_ACCESS_POINT",
    "submissionType": "ANNUAL_SUSTAINABILITY_STATEMENT",
    "deadlineDate": "2025-04-30T00:00:00Z",
    "testMode": false
  },
  "authorizedBy": {
    "userId": "user-cfo-john-smith",
    "role": "CHIEF_FINANCIAL_OFFICER",
    "digitalSignature": "base64_encoded_signature",
    "timestamp": "2025-09-24T18:00:00Z"
  },
  "submissionMetadata": {
    "lei": "LEI1234567890ABCDEF",
    "fiscalYear": "2024",
    "preparationDate": "2025-09-24T00:00:00Z",
    "assuranceLevel": "LIMITED",
    "assuranceProvider": "PwC Dubai"
  }
}
```

Response (202 Accepted):

```
json
{
  "submissionJobId": "submission-job-abc456",
  "status": "PROCESSING",
  "regulatoryPortal": "ESMA_SINGLE_ACCESS_POINT",
  "submissionReference": "ESMA-2025-ORG123-CSRD-001",
  "estimatedProcessingTime": "5-10 minutes",
  "validationSteps": [
    "DIGITAL_SIGNATURE_VERIFICATION",
    "SCHEMA_VALIDATION",
    "BUSINESS_RULES_CHECK",
    "PORTAL_SUBMISSION"
  ],
  "statusUrl": "/submit/regulatory/status/submission-job-abc456"
}
```

GET /submit/regulatory/status/{jobId}

Purpose: Check regulatory submission status

Response (200 OK):

json

```
{
  "submissionJobId": "submission-job-abc456",
  "status": "COMPLETED",
  "submissionResults": {
    "submissionSuccess": true,
    "regulatoryReference": "ESMA-2025-ORG123-CSR-001",
    "submissionTimestamp": "2025-09-24T18:12:45Z",
    "acknowledgmentReceipt": {
      "receiptId": "RCP-ESMA-456789",
      "downloadUrl": "/submissions/receipts/RCP-ESMA-456789.pdf"
    }
  },
  "validationSteps": [
    {
      "step": "DIGITAL_SIGNATURE_VERIFICATION",
      "status": "PASSED",
      "completedAt": "2025-09-24T18:05:15Z"
    },
    {
      "step": "SCHEMA_VALIDATION",
      "status": "PASSED",
      "completedAt": "2025-09-24T18:08:30Z"
    },
    {
      "step": "PORTAL_SUBMISSION",
      "status": "PASSED",
      "completedAt": "2025-09-24T18:12:45Z"
    }
  ],
  "compliance": {
    "deadlineMet": true,
    "daysBeforeDeadline": 218,
    "submissionComplete": true
  }
}
```

2.7 Audit Trail & Documentation

GET /audit/reports

Purpose: Retrieve audit trail for report generation and approval activities

Query Parameters:

- `organizationId` (required): Organization filter
- `reportJobId`: Specific report job
- `standard`: Filter by regulatory standard
- `dateFrom`: Start date filter
- `dateTo`: End date filter
- `action`: Filter by action type
- `userId`: Filter by user actions

Response (200 OK):

json

```
{
  "auditTrail": [
    {
      "auditId": "audit-001",
      "timestamp": "2025-09-24T16:00:00Z",
      "action": "REPORT_GENERATION_INITIATED",
      "userId": "user-sarah-chen",
      "userRole": "ESG_ANALYST",
      "resourceType": "COMPLIANCE_REPORT",
      "resourceId": "report-job-abc123",
      "details": {
        "standard": "CSRD",
        "template": "ESRS_E1_CLIMATE_CHANGE",
        "reportingPeriod": "2024"
      },
      "ipAddress": "10.0.1.45",
      "sessionId": "session-789def"
    },
    {
      "auditId": "audit-002",
      "timestamp": "2025-09-24T17:15:00Z",
      "action": "REPORT_APPROVED",
      "userId": "user-sarah-chen",
      "userRole": "ESG_ANALYST",
      "resourceType": "APPROVAL_WORKFLOW",
      "resourceId": "approval-job-789ghi",
      "details": {
        "approvalLevel": 1,
        "comments": "Report meets CSRD requirements",
        "nextApprover": "user-michael-brown"
      }
    }
  ],
  "pagination": {
    "page": 1,
    "pageSize": 25,
    "totalRecords": 156,
    "totalPages": 7
  }
}
```

3. Error Handling

3.1 Standard Error Response Format

```
json
{
  "error": {
    "code": "COMPLIANCE_VALIDATION_FAILED",
    "message": "Report validation failed due to missing mandatory disclosures",
    "details": "ESRS E1-5 energy consumption data incomplete",
    "timestamp": "2025-09-24T16:30:00Z",
    "requestId": "req-compliance-123abc",
    "path": "/v1/compliance-reporting/validate/compliance",
    "method": "POST",
    "complianceInfo": {
      "standard": "CSRD",
      "template": "ESRS_E1_CLIMATE_CHANGE",
      "missingRequirements": 2,
      "complianceScore": 87.3
    }
  }
}
```

3.2 Compliance-Specific Error Codes

Status Code	Error Code	Description
400	INVALID_STANDARD	Regulatory standard not supported
400	INVALID_TEMPLATE	Report template not found or outdated
400	INVALID_REPORTING_PERIOD	Reporting period format or range invalid
422	COMPLIANCE_VALIDATION_FAILED	Data doesn't meet regulatory requirements
422	INSUFFICIENT_DATA	Not enough data for report generation
422	APPROVAL_WORKFLOW_ERROR	Error in approval process execution
422	REGULATORY_SUBMISSION_FAILED	Submission to regulatory portal failed
424	TEMPLATE_OUTDATED	Template version no longer compliant
503	REGULATORY_PORTAL_UNAVAILABLE	External regulatory system unavailable

4. Performance & SLA

4.1 Processing Time Targets

Operation Type	Target Processing Time	SLA Percentile
Report Generation	< 20 minutes	90th percentile
Compliance Validation	< 2 minutes	95th percentile
Template Retrieval	< 5 seconds	95th percentile
Approval Actions	< 10 seconds	99th percentile
Regulatory Submission	< 10 minutes	90th percentile

4.2 Report Quality Standards

- **Template Accuracy:** 100% compliance with latest regulatory versions
- **Data Completeness:** ≥ 95% for mandatory disclosures
- **Validation Accuracy:** ≥ 99.5% correct identification of compliance gaps
- **Format Consistency:** 100% adherence to regulatory formatting requirements

4.3 Scalability Limits

- **Concurrent Report Generation:** 50 per organization
- **Max Report Size:** 500MB per output file
- **Data Points per Report:** Up to 50,000 metrics
- **Approval Workflow Participants:** Up to 20 approvers per workflow

5. Integration Examples

5.1 Java Spring Boot Client

```
java
```

@Service

```
public class ComplianceReportingClient {

    private final WebClient webClient;
    private final String baseUrl = "https://api.sweep.com/v1/compliance-reporting";

    public ComplianceReportingClient(WebClient.Builder webClientBuilder) {
        this.webClient = webClientBuilder.baseUrl(baseUrl).build();
    }

    public Mono<ReportGenerationResponse> generateCSRDRport(
        String organizationId,
        String template,
        LocalDate startDate,
        LocalDate endDate) {

        ReportGenerationRequest request = ReportGenerationRequest.builder()
            .reportJobId("report-job-" + UUID.randomUUID())
            .organizationId(organizationId)
            .reportConfig(ReportConfig.builder()
                .standard("CSR")
                .template(template)
                .reportingPeriod(ReportingPeriod.builder()
                    .startDate(startDate.atStartOfDay().atZone(ZoneOffset.UTC))
                    .endDate(endDate.atTime(LocalTime.MAX).atZone(ZoneOffset.UTC))
                    .build())
                .outputFormats(List.of("PDF", "XLSX"))
                .build())
            .build();

        return webClient.post()
            .uri("/reports/generate")
            .header("Authorization", "Bearer " + getAccessToken())
            .bodyValue(request)
            .retrieve()
            .bodyToMono(ReportGenerationResponse.class);
    }

    public Mono<ComplianceValidationResponse> validateCompliance(
        String organizationId,
        String standard,
        String template,
        LocalDate startDate,
        LocalDate endDate) {

        ComplianceValidationRequest request = ComplianceValidationRequest.builder()
```

```

        .validationJobId("validation-job-" + UUID.randomUUID())
        .organizationId(organizationId)
        .validationConfig(ValidationConfig.builder()
            .standard(standard)
            .template(template)
            .strictMode(true)
            .includeWarnings(true)
            .build())
        .build();

    return webClient.post()
        .uri("/validate/compliance")
        .header("Authorization", "Bearer " + getAccessToken())
        .bodyValue(request)
        .retrieve()
        .bodyToMono(ComplianceValidationResponse.class);
}

public Mono<ReportStatus> pollReportStatus(String reportJobId) {
    return webClient.get()
        .uri("/reports/generate/status/{jobId}", reportJobId)
        .header("Authorization", "Bearer " + getAccessToken())
        .retrieve()
        .bodyToMono(ReportStatus.class);
}

private String getAccessToken() {
    // Implementation for OAuth token retrieval
    return tokenService.getValidAccessToken();
}
}

// Usage Example
@RestController
public class ReportController {

    @Autowired
    private ComplianceReportingClient complianceClient;

    @PostMapping("/generate-csrd-report")
    public Mono<ResponseEntity<ReportGenerationResponse>> generateReport(
        @RequestBody ReportRequest request) {

        return complianceClient.generateCSRDRpt(
            request.getOrganizationId(),
            "ESRS_E1_CLIMATE_CHANGE",
            request.getStartDate(),

```



```
        request.getEndDate()
    ).map(ResponseEntity::ok);
}

@GetMapping("/report-status/{jobId}")
public Mono<ResponseEntity<ReportStatus>> getReportStatus(@PathVariable String jobId) {
    return complianceClient.pollReportStatus(jobId)
        .map(ResponseEntity::ok);
}
}
```

5.2 Python Integration Example

```
python
```

```
import requests
import time
from datetime import datetime, timedelta
from typing import Dict, List, Optional

class ComplianceReportingClient:
    def __init__(self, access_token: str, base_url: str = "https://api.sweep.com/v1/compliance-reporting"):
        self.base_url = base_url
        self.headers = {
            'Authorization': f'Bearer {access_token}',
            'Content-Type': 'application/json'
        }

    def generate_report(self,
                       organization_id: str,
                       standard: str,
                       template: str,
                       start_date: datetime,
                       end_date: datetime,
                       output_formats: List[str] = None) -> Dict:
        """Generate compliance report"""

        if output_formats is None:
            output_formats = ["PDF", "XLSX"]

        payload = {
            'reportJobId': f'report-job-{int(time.time())}',
            'organizationId': organization_id,
            'reportConfig': {
                'standard': standard,
                'template': template,
                'reportingPeriod': {
                    'startDate': start_date.isoformat() + 'Z',
                    'endDate': end_date.isoformat() + 'Z',
                    'fiscalYear': str(start_date.year)
                },
            },
            'outputFormats': output_formats,
            'language': 'EN',
            'currency': 'USD'
        },
        'dataFilters': {
            'includeEstimatedData': True,
            'qualityThreshold': 85.0,
            'includeThirdPartyData': True
        }
    }
```

```
response = requests.post(  
    f'{self.base_
```